1975

Approximate analytical evaluation of extended-Kalman filters.

Tasdelen, Hasan FOner

Monterey, California. Naval Postgraduate School

APPROXIMATE ANALYTICAL EVALUATION OF EXTENDED-KALMAN FILTERS

Hasan Öner Tasdelen

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

APPROXIMATE ANALYTICAL EVALUATION
OF EXTENDED-KALMAN FILTERS

by

Hasan Öner Taşdelen

December 1975

Thesis Advisor: D. E. Kirk

Approved for public release; distribution unlimited.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Approximate Analytical Evaluation of Extended-Kalman Filters		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; December 1975
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Hasan Oner Tasdelen		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE December 1975
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

**Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Analytic equations, extended-Kalman filter, performance evaluation, Monte-Carlo simulation, conditional expectation.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Analytical equations derived for evaluating linear estimators are applied to extended-Kalman filters for approximate performance evaluation. Two cases were considered, a single known target trajectory and multiple target trajectories with given probabilities of occurrence. For the multiple-trajectory case, equations are derived for the mean and covariance of estimation error in terms of the conditional expectations. Two examples

are presented to compare the use of the analytical equations with Monte-Carlo simulation.

Approximate Analytical Evaluation
of
Extended-Kalman Filters

by

Hasan Öner Taşdelen
Lieutenant, Turkish Navy

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1975

ABSTRACT

Analytical equations derived for evaluating linear estimators are applied to extended-Kalman filters for approximate performance evaluation. Two cases were considered, a single known target trajectory and multiple target trajectories with given probabilities of occurrence. For the multiple-trajectory case, equations are derived for the mean and covariance of estimation error in terms of the conditional expectations. Two examples are presented to compare the use of the analytical equations with Monte-Carlo simulation.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENT

The author is deeply indebted to the United States Navy and the Turkish Navy for providing the opportunity to pursue this work.

I wish to thank my thesis advisor Professor D. E. Kirk and all my professors, for their guidance and counsel.

I. INTRODUCTION

In state estimation problems one of the tasks is the simulation of a filter or estimator after designing it. The purpose of the simulation is to determine the performance of the filter under actual operating conditions and to investigate sensitivity to inaccuracies or approximations present in the design assumptions.

Monte-Carlo simulation is a common simulation algorithm which is currently used. An important drawback of the Monte-Carlo approach is the large amounts of computer time generally required to achieve a reasonable degree of accuracy. To reduce the computation time requirements, fewer Monte-Carlo runs maybe used with an attendant loss of accuracy. Because of these limitations of the Monte-Carlo approach, an alternative method, cosisting of a set of analytical equations, has been derived and a computer algorithm has been established for the evaluation of estimation and prediction error statistics for linear filters. It is possible to characterize the propagation of the means and covariances of estimation error of a filter by difference equations. These difference equations can easily be solved using relatively small amounts of computer time. The only disadvantage of the analytical equation approach is that it only applies to linear filters with precomputed gain schedules.

In practice, one may be confronted by nonlinear problems, such as space vehicle re-entry or orbit determination problems, or fire-control estimation problems. A common approach for nonlinear problems is to use an extended Kalman filter and to evaluate its performance by Monte-Carlo simulation.

The main objective of this research is to investigate the possibility of approximate evaluation of the performance of extended Kalman filters by applying the analytical equation approach. In Chapter II of this thesis the analytical equations are derived for linear time-invariant estimators. In Chapter III, the application of the analytical equations to problems with multiple tracks occurring with given probabilities is investigated and an example is presented. In Chapter IV, a common filtering approach for nonlinear systems is summarized. In Chapter V, the application of the analytical equations to extended Kalman filters is discussed for both the single and multiple track cases by using a nonlinear example. Chapter VI contains simulation results from another example. The simulations were performed using both the analytical equations and the Monte-Carlo algorithm for comparison.

The computer programs that were used are given in Appendix A.

II. ANALYTICAL EQUATIONS

A. ASSUMPTIONS

In the development of analytical equations the following assumptions have been made [1] :

1. The true state trajectory $\underline{X}(k)$ for $k=0,1,2,\ldots$ is known.

2. The measurement equation is

$$\underline{Z}(k)=\underline{h}(\underline{X}(k)) + \underline{V}(k) \tag{2.1}$$

where

$\underline{Z}(k)$ is the q-dimensional measurement vector at time $t=kT$.

$\underline{h}$ is a function (which may be nonlinear) of $\underline{X}(k)$.

$\underline{V}(k)$ is the measurement noise vector with the assumptions

(a). $E\left[\underline{V}(k)\right] =\underline{0}$ for all $k \geq 0$.

(The expected value of the measurement noise is zero for all time)

(b). $E\left[\underline{V}(k)\underline{V}(j)^T\right] =\begin{cases}\underset{\sim}{R}(k) & \text{for } k=j \\ \underset{\sim}{0} & \text{for } k\neq j\end{cases}$ for all $k,j \geq 0$

(the measurement noise is uncorrelated).

(c). Since the target trajectory is known

$E\left[\underline{X}(j)\underline{V}(k)^T\right] =\underline{X}(j).E\left[\underline{V}^T(k)\right] =\underline{0}$ for all $k,j\geq 0$

3. The estimator is linear and described by the equation

$$\hat{\underline{X}}(k/k)=\hat{\underline{X}}(k/k-1)+\underset{\sim}{G}(k)\left[\underline{Z}(k)-\underset{\sim}{C}\hat{\underline{X}}(k/k-1)\right] \tag{2.2}$$

The prediction equation is

$$\hat{\underline{X}}(k/k-1)=\underset{\sim}{\emptyset}\ \hat{\underline{X}}(k-1/k-1)+\underset{\sim}{\Delta}\ \underline{U}(k-1) \qquad (2.3)$$

where

$\hat{\underline{X}}(k/k)$ is the n-dimensional estimate of $\underline{X}(k)$ given measurements

$\underline{Z}(0),\underline{Z}(1),\ldots,\underline{Z}(k)$.

$\hat{\underline{X}}(k/k-1)$ is the predicted value of $\underline{X}(k)$ given measurements $\underline{Z}(0)$, $\underline{Z}(1),\ldots,\underline{Z}(k-1)$.

$\underset{\sim}{\emptyset}$ and $\underset{\sim}{C}$ are nxn and qxn matrices, respectively, and they are assumed to be known.

$\underline{U}(k-1)$ is the m-dimensional deterministic forcing vector at time $t=(k-1)T$.

$\underset{\sim}{\Delta}$ is an nxm known matrix.

$\underset{\sim}{G}(k)$ is the nxm gain matrix.

The gains may be found by any method. For example, optimal estimation gains which minimize the sum of the variance of estimation error can be obtained by using the Kalman equations:

$$\underset{\sim}{G}(k)=\underset{\sim}{P}(k/k-1)\ \underset{\sim}{C}^T \left[\ \underset{\sim}{C}\ \underset{\sim}{P}(k/k-1)\ \underset{\sim}{C}^T+\underset{\sim}{R}(k)\ \right]^{-1} \qquad (2.4)$$

$$\underset{\sim}{P}(k/k)=\left[\underset{\sim}{I}-\underset{\sim}{G}(k)\ \underset{\sim}{C}\right]\ \underset{\sim}{P}(k/k-1) \qquad (2.5)$$

$$\underset{\sim}{P}(k/k-1)=\underset{\sim}{\emptyset}\ \underset{\sim}{P}(k-1/k-1)\ \underset{\sim}{\emptyset}^T+\underset{\sim}{Q}(k-1) \qquad (2.6)$$

with the initial conditions

$$\underset{\sim}{P}(0/-1)=\underset{\sim}{P}o=E\left\{\left[\underline{X}(0)-\underline{X}_o\right]\cdot\left[\underline{X}(0)-\underline{X}_o\right]^T\right\}$$

$$\hat{\underline{X}}(0/-1)=\bar{\underline{X}}_o$$

(In this research gains determined by using the Kalman equations have been used for simulations.)

Where

$$\underset{\sim}{P}(k/k)=E\left\{\left[\hat{\underline{X}}(k/k) - \underline{X}(k)\right] \cdot \left[\hat{\underline{X}}(k/k) - \underline{X}(k)\right]^T\right\} \quad (2.7)$$

which is the covariance matrix of theoretical estimation error, and

$$\underline{P}(k/k-1)=E\left\{\left[\hat{\underline{X}}(k/k-1)-\underline{X}(k)\right] \cdot\left[\hat{\underline{X}}(k/k-1)-\underline{X}(k)\right]^T\right\} \quad (2.8)$$

$\underline{Q}(k)$ is the covariance matrix of the random forcing input $\underline{W}(k)$. i.e.

$$E\left[\underline{W}(k)\ \underline{W}(j)^T\right]=\begin{cases}\underline{Q}(k) & \text{for } k=j\\ \underline{0} & \text{for } k\neq j\end{cases} \quad (2.9)$$

If the estimation equation (2.2) is initialized with the value

$$\hat{\underline{X}}(0/-1) = \underline{X}_0 = E\ \ \underline{X}(0) \quad (2.10)$$

it can be shown that the optimal estimate $\hat{\underline{X}}(k/k)$ is unbiased, i.e.

$$E\left[\hat{\underline{X}}(k/k) - \underline{X}(k)\right] =\underline{0} \quad (2.11)$$

With the assumptions given above one can derive difference equations for the mean of estimation error, the covariance of estimation error, the mean of N-step prediction error and the covariance of N-step prediction error. These equations are derived for the time-invariant estimator given by equations (2.2) and (2.3) in the following section.

B. DERIVATION OF ANALYTICAL EQUATIONS

1. Mean and covariance of estimation error

The true state of the target at time k is $\underline{X}(k)$; thus, using Equations (2.2) and (2.3) the estimation error $\tilde{\underline{X}}(k)$ at time k is

$$\underline{\tilde{X}}(k) \triangleq \underline{\hat{X}}(k/k) - \underline{X}(k)$$

$$= \underset{\sim}{\phi}\,\underline{\hat{X}}(k-1/k-1) + \underset{\sim}{\Delta}\,\underline{U}(k-1) + \underset{\sim}{G}(k)\Big[\,\underline{Z}(k) - \underset{\sim}{C}\,\underset{\sim}{\phi}\,\underline{X}(k-1/k-1) - \underset{\sim}{C}\,\underset{\sim}{\Delta}\,\underline{U}(k-1)\Big] - \underline{X}(k) \qquad (2.12)$$

Substituting Equation (2.1) into (2.12) gives

$$\underline{\tilde{X}}(k) = \Big[\underset{\sim}{I} - \underset{\sim}{G}(k)\,\underset{\sim}{C}\Big]\,\underset{\sim}{\phi}\,\underline{\hat{X}}(k-1/k-1) + \underset{\sim}{G}(k)\,\underline{V}(k) + \underset{\sim}{G}(k)\,\underline{h}(\underline{X}(k)) - \underline{X}(k) + \Big[\underset{\sim}{I} - \underset{\sim}{G}(k)\,\underset{\sim}{C}\Big]\underset{\sim}{\Delta}\,\underline{U}(k-1) \qquad (2.13)$$

By defining the deterministic matrices S(k) and D(k) as

$$\underset{\sim}{S}(k) = \Big[\underset{\sim}{I} - \underset{\sim}{G}(k)\,\underset{\sim}{C}\Big]\,\underset{\sim}{\phi} \qquad (2.14)$$

$$\underset{\sim}{D}(k) = \Big[\underset{\sim}{I} - \underset{\sim}{G}(k)\,\underset{\sim}{C}\Big]\underset{\sim}{\Delta}\,\underline{U}(k-1) + \underset{\sim}{G}(k)\,\underline{h}\,(\underline{X}(k)) - \underline{X}(k) \qquad (2.15)$$

Equation (2.13) becomes

$$\underline{\tilde{X}}(k) = \underset{\sim}{S}(k)\,\underline{\hat{X}}(k-1/k-1) + \underset{\sim}{D}(k) + \underset{\sim}{G}(k)\,\underline{V}(k) \qquad (2.16)$$

The mean of estimation error is defined as

$$E\Big[\underline{\tilde{X}}(k)\Big] = E\Big[\,\underset{\sim}{S}(k)\,\underline{\hat{X}}(k-1/k-1) + \underset{\sim}{D}(k) + \underset{\sim}{G}(k)\,\underline{V}(k)\Big]$$

The matrices $\underset{\sim}{S}(k)$ and $\underset{\sim}{D}(k)$ are deterministic, hence

$$E\Big[\,\underset{\sim}{G}(k)\,\underline{V}(k)\,\Big] = \underset{\sim}{G}(k)\,E\Big[\,\underline{V}(k)\Big] = \underline{0}$$

and using properties of the expectation operator gives

$$E\Big[\,\underline{\tilde{X}}(k)\Big] = \underset{\sim}{S}(k)\,E\Big[\,\underline{\hat{X}}(k-1/k-1)\,\Big] + \underset{\sim}{D}(k)$$

Defining

$$\underline{\tilde{\mu}}(k) = E\Big[\,\underline{\tilde{X}}(k)\Big]$$

gives

$$\underline{\tilde{\mu}}(k) = \underset{\sim}{S}(k)\,E\Big[\,\underline{\hat{X}}(k-1/k-1)\,\Big] + \underset{\sim}{D}(k) \qquad (2.17)$$

Equation (2.16) can also be written as

$$\begin{aligned}\tilde{\underline{X}}(k) &= \underset{\sim}{S}(k)\,\hat{\underline{X}}(k-1/k-1) + \underset{\sim}{D}(k) + \underset{\sim}{G}(k)\,\underline{V}(k) \\ &\quad + \left[\underset{\sim}{S}(k)\,\underline{X}(k-1) - \underset{\sim}{S}(k)\,\underline{X}(k-1)\right] \\ &= \underset{\sim}{S}(k)\left[\hat{\underline{X}}(k-1/k-1) - \underline{X}(k-1)\right] + \underset{\sim}{D}(k) \\ &\quad + \underset{\sim}{G}(k)\,\underline{V}(k) + \underset{\sim}{S}(k)\,\underline{X}(k-1)\end{aligned}$$

or

$$\begin{aligned}\tilde{\underline{X}}(k) &= \underset{\sim}{S}(k)\,\tilde{\underline{X}}(k-1) + \underset{\sim}{D}(k) + \underset{\sim}{G}(k)\,\underline{V}(k) \\ &\quad + \underset{\sim}{S}(k)\,\underline{X}(k-1)\end{aligned} \tag{2.18}$$

Since $\underline{X}(k-1)$ is deterministic and

$$E\left[\tilde{\underline{X}}(k-1)\right] = \tilde{\underline{\mu}}(k-1)$$

Equation (2.17) can be written as

$$\tilde{\underline{\mu}}(k) = \underset{\sim}{S}(k)\,\tilde{\underline{\mu}}(k-1) + \underset{\sim}{D}(k) + \underset{\sim}{S}(k)\,\underline{X}(k-1) \tag{2.19}$$

Equation (2.19) defines the mean of estimation error at time k, in terms of the mean of estimation error at time (k-1).

The covariance of estimation error is defined as

$$\begin{aligned}\underset{\sim}{\tilde{P}}(k/k) &= E\left\{\left[\tilde{\underline{X}}(k) - \tilde{\underline{\mu}}(k)\right]\cdot\left[\tilde{\underline{X}}(k) - \tilde{\underline{\mu}}(k)\right]^T\right\} \\ &= E\left[\underline{Y}(k)\,\underline{Y}(k)^T\right]\end{aligned} \tag{2.20}$$

where

$$\underline{Y}(k) = \tilde{\underline{X}}(k) - \tilde{\underline{\mu}}(k) \tag{2.21}$$

One can obtain a difference equation for $\underline{Y}(k)$ by substituting (2.18) and (2.19) into (2.21), that is,

$$\begin{aligned}
\underline{Y}(k) &= \underset{\sim}{S}(k)\ \tilde{\underline{X}}(k-1) + \cancel{\underline{D}(k)} + \underset{\sim}{G}(k)\ \underline{V}(k) + \cancel{\underset{\sim}{S}(k)\ \underline{X}(k-1)} \\
&\quad - \underset{\sim}{S}(k)\ \tilde{\underline{\mu}}(k-1) - \cancel{\underline{D}(k)} - \cancel{\underset{\sim}{S}(k)\ \underline{X}(k-1)} \\
&= \underset{\sim}{S}(k) \left[\tilde{\underline{X}}(k-1) - \tilde{\underline{\mu}}(k-1) \right] + \underset{\sim}{G}(k)\ \underline{V}(k) \\
&= \underset{\sim}{S}(k)\ \underline{Y}(k-1) + \underset{\sim}{G}(k)\ \underline{V}(k) \qquad (2.22)
\end{aligned}$$

Then

$$\begin{aligned}
\tilde{\underset{\sim}{P}}(k/k) &= E\left[\underline{Y}(k)\ \underline{Y}(k)^T\right] \\
&= E\left[\underset{\sim}{S}(k)\ \underline{Y}(k-1)\ \underline{Y}(k-1)^T\ \underset{\sim}{S}(k)^T\right] \\
&\quad + E\left[\underset{\sim}{S}(k)\ \underline{Y}(k-1)\ \underline{V}(k)^T\ \underset{\sim}{G}(k)^T\right] \\
&\quad + E\left[\underset{\sim}{G}(k)\ \underline{V}(k)\ \underline{Y}(k-1)^T\ \underset{\sim}{S}(k)^T\right] \\
&\quad + E\left[\underset{\sim}{G}(k)\ \underline{V}(k)\ \underline{V}(k)^T\ \underset{\sim}{G}(k)^T\right]
\end{aligned}$$

Using properties of the expectation operator yields

$$\begin{aligned}
\tilde{\underset{\sim}{P}}(k/k) &= \underset{\sim}{S}(k)\ E\left[\underline{Y}(k-1)\ \underline{Y}(k-1)^T\right] \underset{\sim}{S}(k)^T \\
&\quad + \underset{\sim}{S}(k)\ E\left[\underline{Y}(k-1)\ \underline{V}(k)^T\right] \underset{\sim}{G}(k)^T \\
&\quad + \underset{\sim}{G}(k)\ E\left[\underline{V}(k)\ \underline{Y}(k-1)^T\right] \underset{\sim}{S}(k)^T \\
&\quad + \underset{\sim}{G}(k)\ \underset{\sim}{R}(k)\ \underset{\sim}{G}(k)^T \qquad (2.23)
\end{aligned}$$

To obtain expressions for $E\left[\underline{Y}(k-1)\ \underline{V}(k)^T\right]$ and $E\left[\underline{V}(k)\ \underline{Y}(k-1)^T\right]$ one can start by deriving an expression for $\underline{Y}(0)$. The definition of $\underline{Y}(0)$ is

$$\begin{aligned}
\underline{Y}(0) &= \tilde{\underline{X}}(0) - \tilde{\underline{\mu}}(0) \\
&= \hat{\underline{X}}(0/0) - \underline{X}(0) - E\left[\tilde{\underline{X}}(0)\right]
\end{aligned}$$

$$= \underline{\hat{X}}(0/0) - \underline{X}(0) - E\left[\underline{\hat{X}}(0/0) - \underline{X}(0)\right]$$

Since $\quad E\left[\underline{X}(0)\right] = \underline{X}(0),$

$$\underline{Y}(0) = \underline{\hat{X}}(0/0) - E\left[\underline{\hat{X}}(0/0)\right] \tag{2.24}$$

But

$$\begin{aligned} \underline{\hat{X}}(0/0) &= \underline{\hat{X}}(0/-1) + \underset{\sim}{G}(0)\left[\underline{Z}(0) - \underset{\sim}{C}\,\underline{\hat{X}}(0/-1)\right] \\ &= \left[\underset{\sim}{I} - \underset{\sim}{G}(0)\,\underset{\sim}{C}\right]\underline{\hat{X}}(0/-1) + \underset{\sim}{G}(0)\,\underline{h}(\underline{X}(0)) \\ &\quad + \underset{\sim}{G}(0)\,\underline{V}(0) \end{aligned} \tag{2.25}$$

Substituting (2.25) into (2.24) gives

$$\begin{aligned} \underline{Y}(0) &= \left[\underset{\sim}{I} - \underset{\sim}{G}(0)\,\underset{\sim}{C}\right]\underline{\hat{X}}(0/-1) + \underset{\sim}{G}(0)\,\underline{h}(\underline{X}(0)) \\ &\quad + \underset{\sim}{G}(0)\,\underline{V}(0) - E\left\{\left[\underset{\sim}{I} - \underset{\sim}{G}(0)\,\underset{\sim}{C}\right]\underline{\hat{X}}(0/-1)\right. \\ &\quad \left. + \underset{\sim}{G}(0)\,\underline{h}(\underline{X}(0)) + \underset{\sim}{G}(0)\,\underline{V}(0)\right\} \end{aligned}$$

However,

$$E\left[\underset{\sim}{G}(0)\,\underline{V}(0)\right] = \underset{\sim}{G}(0)\,E\left[\underline{V}(0)\right] = 0$$

$$E\left[\underset{\sim}{G}(0)\,\underline{h}(\underline{X}(0))\right] = \underset{\sim}{G}(0)\,\underline{h}(\underline{X}(0))$$

and using properties of the expectation operator gives

$$\begin{aligned} \underline{Y}(0) &= \left[\underset{\sim}{I} - \underset{\sim}{G}(0)\,\underset{\sim}{C}\right]\underline{\hat{X}}(0/-1) + \cancel{\underset{\sim}{G}(0)\,\underline{h}(\underline{X}(0))} \\ &\quad + \underset{\sim}{G}(0)\,\underline{V}(0) - \left[\underset{\sim}{I} - \underset{\sim}{G}(0)\,\underset{\sim}{C}\right] E\left[\underline{\hat{X}}(0/-1)\right] \\ &\quad - \cancel{\underset{\sim}{G}(0)\,\underline{h}(\underline{X}(0))} \end{aligned}$$

But $\underline{\hat{X}}(0/-1)$ is a deterministic, known quantity, so

$$E\left[\underline{\hat{X}}(0/-1)\right] = \underline{\hat{X}}(0/-1) \text{ and}$$

$$\underline{Y}(0) = \underset{\sim}{G}(0)\,\underline{V}(0) \tag{2.26}$$

Using Equation (2.22)

$$\underline{Y}(1) = \underset{\sim}{S}(1)\ \underline{Y}(0) + \underset{\sim}{G}(1)\ \underline{V}(1)$$

$$= \underset{\sim}{S}(1)\ \underset{\sim}{G}(0)\ \underline{V}(0) + \underset{\sim}{G}(1)\ \underline{V}(1) \qquad (2.27)$$

and it can be shown that

$$\underline{Y}(k) = \sum_{j=0}^{k-1} \left[\prod_{i=0}^{k-j-1} \underset{\sim}{S}(k-i) \right] \underset{\sim}{G}(j)\ \underline{V}(j) + \underset{\sim}{G}(k)\ \underline{V}(k) \qquad (2.28)$$

Equation (2.28) shows that $\underline{Y}(k)$ is a linear combination of $\underline{V}(k)$ with coefficients which are known constant matrices, i.e.

$$\underline{Y}(k) = \sum_{\ell=0}^{k} \underset{\sim}{L}_{\ell}(k)\ \underline{V}(\ell) \qquad (2.29)$$

where

$$\underset{\sim}{L}_{\ell}(k) = \sum_{i=0}^{k-\ell-1} \underset{\sim}{S}(k-i)\ G(\ell) \quad \text{for } l=0,1,2,\ldots,(k-1)$$

$$L_k(k) = \underset{\sim}{G}(k) \qquad (2.30)$$

From Equation (2.29) it is easily seen that

$$E\left[\underline{Y}(k-1)\ \underline{V}(k)^T\right] = E\left[\underline{V}(k)\ \underline{Y}(k-1)^T\right] = \underline{0}$$

Because of the assumption that $E\left[\underline{V}(k)\ \underline{V}(j)^T\right] = \underline{0}$ for $k \neq j$.

This result reduces Equation (2.23) to

$$\underset{\sim}{\tilde{P}}(k/k) = \underset{\sim}{S}(k)\ \underset{\sim}{\tilde{P}}(k-1/k-1)\ \underset{\sim}{S}(k)^T$$

$$+ \underset{\sim}{G}(k)\ R(k)\ \underset{\sim}{G}(k)^T \qquad (2.31)$$

Equation (2.31) is the error covariance propogation equation which expresses the covariance of estimation error at time k in terms of the covariance of estimation error at time (k-1).

2. Initial conditions

To use Equations (2.19) and (2.31) one needs to know initial condition values.

The mean of estimation error at time k=0 is

$$\tilde{\underline{\mu}}(0) = E\left[\hat{\underline{X}}(0/0) - \underline{X}(0)\right]$$
$$= E\left[\hat{\underline{X}}(0/0)\right] - \underline{X}(0)$$

Using Equation (2.25) this becomes

$$\tilde{\underline{\mu}}(0) = E\left\{\left[\underset{\sim}{I} - \underset{\sim}{G}(0)\ \underset{\sim}{C}\right] \hat{\underline{X}}(0/-1) + \underset{\sim}{G}(0)\ \underline{h}(\underline{X}(0)) + \underset{\sim}{G}(0)\ \underline{V}(0)\right\} - \underline{X}(0)$$

But

$$E\left[\hat{\underline{X}}(0/-1)\right] = \hat{\underline{X}}(0/-1)$$

$$E\left[\underset{\sim}{G}(0)\ \underline{h}(\underline{X}(0))\right] = \underset{\sim}{G}(0)\ \underline{h}(\underline{x}(0))$$

and $$E\left[\underset{\sim}{G}(0)\ \underline{V}(0)\right] = 0$$

thus,

$$\tilde{\underline{\mu}}(0) = \left[\underset{\sim}{I} - \underset{\sim}{G}(0)\ \underset{\sim}{C}\right] \hat{\underline{X}}(0/-1) - \underline{X}(0) + \underset{\sim}{G}(0)\ \underline{h}(\underline{x}(0)) \qquad (2.32)$$

Equation (2.32) gives the initial condition for Equation (2.19).

The covariance of estimation error at time k=0 is

$$\underset{\sim}{\tilde{P}}(0/0) = E\left[\underline{Y}(0)\ \underline{Y}(0)^T \right] \tag{2.33}$$

Substituting Equation (2.26) into (2.33) gives

$$\underset{\sim}{\tilde{P}}(0/0) = E\left[\underset{\sim}{G}(0)\ \underline{V}(0)\ \underline{V}(0)^T\ \underset{\sim}{G}(0)^T \right]$$

$$= \underset{\sim}{G}(0)\ \underset{\sim}{R}(0)\ \underset{\sim}{G}(0)^T \tag{2.35}$$

Equation (2.35) gives the initial condition for equation (2.31).

3. Mean and covariance of prediction error

The one-step prediction is given by

$$\hat{\underline{X}}(k+1/k) = \underset{\sim}{\phi}\hat{\underline{X}}(k/k) + \underset{\sim}{\Delta}\underline{U}(k) \tag{2.36}$$

and the N-step prediction, based on the estimate $\underline{X}(k/k)$ is

$$\hat{\underline{X}}_p(k+N/k) = \underset{\sim}{\phi}^N\ \underline{X}(k/k) + \sum_{i=0}^{N-1} \underset{\sim}{\phi}^{N-i-1}\underset{\sim}{\Delta}\ \underline{U}(k+i) \tag{2.37}$$

Defining the deterministic matrix $\underset{\sim}{A}(N)$ as

$$\underset{\sim}{A}(N) = \sum_{i=0}^{N-1} \underset{\sim}{\phi}^{N-i-1}\underset{\sim}{\Delta}\ \underline{U}(k+i) \tag{2.38}$$

then

$$\hat{\underline{X}}_p(k+N/k) = \underset{\sim}{\phi}^N\ \hat{\underline{X}}(k/k) + \underset{\sim}{A}(N) \tag{2.39}$$

The N-step prediction error is

$$\tilde{\underline{X}}_p(k+N/k) = \hat{\underline{X}}_p(k+N/k) - \underline{X}(k+N) \tag{2.40}$$

$$= \underset{\sim}{\phi}^N\ \hat{\underline{X}}(k/k) + \underset{\sim}{A}(N) - \underline{X}(k+N)$$

The mean of the N-step prediction error is defined as

$$\tilde{\underline{\mu}}_p(k+N/k) = E\left[\tilde{\underline{X}}_p(k+N/k)\right]$$

$$= E\left[\underset{\sim}{\phi}^N\,\hat{\underline{X}}(k/k) + \underset{\sim}{A}(N) - \underline{X}(k+N)\right]$$

$$= \underset{\sim}{\phi}^N\,E\left[\hat{\underline{X}}(k/k)\right] + \underset{\sim}{A}(N) - \underline{X}(k+N) \qquad (2.41)$$

But

$$\tilde{\underline{\mu}}(k) = E\left[\hat{\underline{X}}(k/k) - \underline{X}(k)\right]$$

$$= E\left[\hat{\underline{X}}(k/k)\right] - \underline{X}(k)$$

and

$$E\left[\hat{\underline{X}}(k/k)\right] = \tilde{\underline{\mu}}(k) + \underline{X}(k) \qquad (2.42)$$

Substituting (2.42) into (2.41) gives

$$\tilde{\underline{\mu}}_p(k+N/k) = \underset{\sim}{\phi}^N\,\tilde{\underline{\mu}}(k) + \underset{\sim}{\phi}^N\,\underline{X}(k) + \underset{\sim}{A}(N) - \underline{X}(k+N) \qquad (2.43)$$

The covariance of N-step prediction error is

$$\underset{\sim}{\tilde{P}}(k+N/k) = E\left\{\left[\tilde{\underline{X}}_p(k+N/k) - \tilde{\underline{\mu}}_p(k+N/k)\right]\cdot\left[\tilde{\underline{X}}_p(k+N/k) - \tilde{\underline{\mu}}_p(k+N/k)\right]^T\right\} \qquad (2.44)$$

Using (2.40) and (2.43) yields

$$\tilde{\underline{X}}_p(k+N/k) - \tilde{\underline{\mu}}_p(k+N/k) = \underset{\sim}{\phi}^N\,\hat{\underline{X}}(k/k) + \cancel{\underset{\sim}{A}(N)} - \cancel{\underline{X}(k+N)}$$

$$- \underset{\sim}{\phi}^N\,\tilde{\underline{\mu}}(k) - \underset{\sim}{\phi}^N\,\underline{X}(k) - \cancel{\underset{\sim}{A}(N)}$$

$$+ \cancel{\underline{X}(k+N)}$$

$$= \underset{\sim}{\phi}^N\left[\hat{\underline{X}}(k/k) - \underline{X}(k) - \tilde{\underline{\mu}}(k)\right]$$

$$= \underset{\sim}{\phi}^N\left[\tilde{\underline{X}}(k/k) - \tilde{\underline{\mu}}(k)\right] \qquad (2.45)$$

which, when substituted into (2.44) gives

$$\tilde{\underset{\sim}{P}}(k+N/k) = E\left\{ \underset{\sim}{\emptyset}^N \left[\tilde{\underline{X}}(k/k) - \tilde{\underset{\sim}{\mu}}(k) \right] \left[\tilde{\underline{X}}(k/k) - \tilde{\underset{\sim}{\mu}}(k) \right]^T \left[\emptyset^N \right]^T \right\}$$

$$= \underset{\sim}{\emptyset}^N \tilde{\underset{\sim}{P}}(k/k) \left[\underset{\sim}{\emptyset}^N \right]^T \qquad (2.46)$$

Equations (2.43) and (2.46) give the mean and covariance of N-step prediction error based on the mean and covariance of estimation error at time k.

In practice, the measurement equations are often linear, i.e.

$$\underline{Z}(k) = \underset{\sim}{C}\, \underline{X}(k) + \underline{V}(k) \qquad (2.47)$$

Then, in the derivation of the analytical equations, $\underline{h}(\underline{X}(k))$ can be replaced with $\underset{\sim}{C}\, \underline{X}(k)$. This will change Equation (2.15) to

$$\underset{\sim}{D}(k) = \left[\underset{\sim}{I} - \underset{\sim}{G}(k)\, \underset{\sim}{C} \right] \left[\underset{\sim}{\Delta} \underline{U}(k-1) - \underline{X}(k) \right] \qquad (2.48)$$

and Equation (2.32) to

$$\tilde{\underset{\sim}{\mu}}(0) = \left[\underset{\sim}{I} - \underset{\sim}{G}(0)\, \underset{\sim}{C} \right] \left[\hat{\underline{X}}(0/-1) - \underline{X}(0) \right] \qquad (2.49)$$

4. Summary of key equations

(1). Mean of estimation error

$$\tilde{\underset{\sim}{\mu}}(k) = \underset{\sim}{S}(k) \tilde{\underset{\sim}{\mu}}(k-1) + \underset{\sim}{D}(k) + \underset{\sim}{S}(k)\, \underline{X}(k-1) \qquad (2.19)$$

(2). Covariance of estimation error

$$\tilde{\underset{\sim}{P}}(k/k) = \underset{\sim}{S}(k)\, \tilde{\underset{\sim}{P}}(k-1/k-1)\, \underset{\sim}{S}(k)^T + \underset{\sim}{G}(k)\, \underset{\sim}{R}(k)\, \underset{\sim}{G}(k)^T \qquad (2.31)$$

where

$$\underset{\sim}{S}(k) = \left[\underset{\sim}{I} - \underset{\sim}{G}(k)\ \underset{\sim}{C}\right] \underset{\sim}{\emptyset} \tag{2.14}$$

$$D(k) = \left[\underset{\sim}{I} - \underset{\sim}{G}(k)\ \underset{\sim}{C}\right]\underset{\sim}{\Delta}\,\underline{U}(k-1)$$

$$+\ \underset{\sim}{G}(k)\underline{h}(\underline{X}(k)) - \underline{X}(k) \tag{2.15}$$

with initial conditions

$$\tilde{\mu}(0)=\left[\underset{\sim}{I} - \underset{\sim}{G}(0)\ \underset{\sim}{C}\right] \hat{\underline{X}}(0/-1) - X(0)$$

$$+G(0)\ \underline{h}(\underline{X}(0)) \tag{2.32}$$

$$\underset{\sim}{\tilde{P}}(0/0) = \underset{\sim}{G}(0)\ \underset{\sim}{R}(0)\ \underset{\sim}{G}(0)^T \tag{2.35}$$

(3). Mean and covariance of N-step prediction error

$$\tilde{\mu}_p(k+N/k) = \underset{\sim}{\emptyset}^N\tilde{\mu}(k) + \underset{\sim}{\emptyset}^N\ \underline{X}(k) + \underset{\sim}{A}(N)$$

$$-\ \underline{X}(k+N) \tag{2.43}$$

$$\underset{\sim}{\tilde{P}}(k+N/k) = \underset{\sim}{\emptyset}^N\ \underset{\sim}{\tilde{P}}(k/k)\left[\underset{\sim}{\emptyset}^N\right]^T \tag{2.46}$$

where

$$\underset{\sim}{A}(N) = \sum_{i=0}^{N-1} \underset{\sim}{\emptyset}^{N-i-1}\ \underset{\sim}{\Delta}\ \underline{U}(k+i) \tag{2.38}$$

If the system is linear and time-varying the $\underset{\sim}{\emptyset}, \underset{\sim}{\Delta}$ and $\underset{\sim}{C}$ matrices are time dependent and are denoted by $\underset{\sim}{\emptyset}(k)$, $\underset{\sim}{\Delta}(k)$, $\underset{\sim}{C}(k)$. Since

$$\hat{\underline{X}}(k/k-1) = \underset{\sim}{\emptyset}(k-1)\ \underline{X}(k-1/k-1) + \underset{\sim}{\Delta}(k-1)\ \underline{U}(k-1) \tag{2.50}$$

and

$$\underline{Z}(k) = \underline{h}(\underline{X}(k)) + \underline{V}(k) \tag{2.51}$$

it is only necessary to replace the $\underset{\sim}{\phi}$, $\underset{\sim}{\Delta}$ and $\underset{\sim}{C}$ matrices with $\underset{\sim}{\phi}(k-1)$, $\underset{\sim}{\Delta}(k-1)$ and $\underset{\sim}{C}(k)$ in Equations (2.14), (2.43), (2.15), (2.46) and (2.38).

The analytical equations derived in this chapter are applicable to linear filters with precomputed gain schedules.

From Equation (2.31) it is seen that the covariance of estimation error is independent of the track, however, the mean of estimation error is track dependent.

III. APPLICATION OF THE ANALYTICAL EQUATIONS TO THE CASE OF MORE THAN ONE TRACK

The equations derived in the previous chapter are based on one known track. Application of the analytical equations using one known track has been studied in Reference [1] and results have been tabulated.

If it is desired to evaluate the performance of a filter for various tracks, a different track can be used for each iteration (ensemble member) in a Monte-Carlo simulation. An alternative approach based on the previously derived analytical equations is:

(1). Apply the analytical equations to calculate the mean and covariance of estimation error for each track individually; these are the conditional means and covariances.

(2). Calculate the overall mean and covariance using the results of (1) and relationships involving the means of conditional expectation.

Assume that there are several tracks to be considered;

$$\overset{(1)}{\underline{X}}(k), \overset{(2)}{\underline{X}}(k), \ldots, \overset{(i)}{\underline{X}}(k), \ldots, \overset{(n)}{\underline{X}}(k)$$

where each track has a given probability of occurrence

$$p_1, p_2, p_3, \ldots, p_i, \ldots p_n \quad , \text{ i.e.}$$

$$p\left[\underline{X}(k) = \overset{(i)}{\underline{X}}(k) \right] = p_i$$

$$p_1 + p_2 + \ldots + p_i + \ldots + p_n = 1 \tag{3.1}$$

The mean $\tilde{\underline{\mu}}^{(i)}(k)$ and the covariance $\tilde{\underset{\sim}{P}}^{(i)}(k/k)$ can be calculated for the i'th track $\underline{X}^{(i)}(k)$. Since $X^{(i)}(k)$ has been used for the calculation of the $\underline{\mu}^{(i)}(k)$, $\tilde{\underline{\mu}}^{(i)}(k)$ is the conditional expectation, i.e.

$$\tilde{\underline{\mu}}^{(i)}(k) = E\left[\tilde{\underline{X}}(k)/\underline{X}(k) = \underline{X}^{(i)}(k)\right] \tag{3.2}$$

Which is the mean of estimation error given the track $\underline{X}^{(i)}(k)$. Thus, conditional expectations can be computed for each of the tracks.

To calculate the mean of estimation error, one needs a relationship between the conditional means and the mean.

A. CONDITIONAL EXPECTATION

The conditional expectation is defined as [2]:

$$E\left[\underline{A}/\underline{B}=\underline{b}_j\right] = \int_{-\infty}^{+\infty}\cdots\int_{-\infty}^{+\infty} \underline{a}\, f_A(\underline{a}\ /\ \underline{B}=\underline{b}_j)\ .da_1\ .da_2 \ldots .da_n \tag{3.3}$$

Where

$\underline{A}$ is a continuous random vector.

$\underline{B}$ is a discrete random vector.

$f(\underline{a}\ /\ \underline{B}=\underline{b}_j)$ is the probability density function of the random vector $\underline{A}$ given $\underline{B}=\underline{b}_j$.

The expected value of a continuous random vector $\underline{A}$ is

$$E\left[\underline{A}\right] = \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} \underline{a}\ f_A(\underline{a})\ da_1 \cdot da_2 \cdot \ldots \ldots da_n \quad (3.4)$$

Equation (3.4) can also be written in terms of the joint probability density function of continuous random vectors, i.e.

$$E\left[\underline{A}\right] = \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} \underline{a}\ f_{AB}(\underline{a},\underline{b}) \cdot da_1 \ldots da_n \cdot db_1 \ldots db_m \quad (3.5)$$

From probability theory the joint probability density function can be expressed in terms of conditional density function. Since the random vector $\underline{B}$ is discrete (point conditioning),

$$f_A(\underline{a} / \underline{B}=\underline{b}_j) = \frac{f_{AB}(\underline{a},\underline{b})}{P\left[\underline{B}=\underline{b}_j\right]} \quad (3.6)$$

or

$$f_{AB}(\underline{a},\underline{b}) = f_A(\underline{a} / \underline{B}=\underline{b}_j) \cdot P\left[\underline{B}=\underline{b}_j\right] \quad (3.7)$$

Substituting Equation (3.7) into (3.5) and replacing the integrals with a summation sign for the random vector $\underline{B}$ (since $\underline{B}$ is a discrete random vector) gives

$$E\left[\underline{A}\right] = \sum_j \left[\int_{-\infty}^{+\infty} \ldots\ldots \int_{-\infty}^{+\infty} \underline{a}\ f_A(\underline{a} / \underline{B}=\underline{b}_j) \cdot da_1 \ldots da_n\right] \cdot P\left[\underline{B}=\underline{b}_j\right] \quad (3.8)$$

The term inside the brackets in Equation (3.8) is the conditional expectation given by Equation (3.3). Thus, Equation (3.8) takes the final form of

$$E\left[A \right] = \sum_{j} E\left[\underline{A} \,/\, \underline{B}=\underline{b}_j \right] \cdot P\left[\underline{B}=\underline{b}_j \right] \tag{3.9}$$

Equation (3.9) can be used to define the mean of estimation error in terms of individual means calculated for each track, i.e.

$$\tilde{\underline{\mu}}(k) = E\left[\tilde{\underline{X}}(k)\right]$$

$$= \sum_{i=1}^{n} E\left[\tilde{\underline{X}}(k) \,/\, \underline{X}(k)=\underline{X}^{(i)}(k) \right] \cdot P\left[\underline{X}(k)=\underline{X}^{(i)}(k) \right] \tag{3.10}$$

$$= \sum_{i=1}^{n} \tilde{\underline{\mu}}^{(i)}(k) \cdot P_i \tag{3.11}$$

where

$$P_i = P\left[\underline{X}(k)=\underline{X}^{(i)}(k) \right]$$

From Equation (2.31) it is seen that the covariance of estimation error is independent of the track for linear systems. So it can be calculated once for all tracks.

B. A LINEAR PROBLEM

An example of the application of the analytical equations to a simplified fire control estimation problem has been

presented in [1]. In this section, the same problem is considered for three tracks with given probabilities of occurrence. The simulation has been done using both the analytical equations (program 'EVAL') and a Monte-Carlo algorithm.

The problem is a part of a simplified anti-aircraft fire control estimation problem in one dimension. The filter is a Kalman filter which is derived by solving Equations (2.2) through (2.11) subject to the following assumptions. The model of target motion is a point mass moving with constant velocity. The state equations are

$$\underline{X}(k+1) = \underset{\sim}{\phi}\, \underline{X}(k) + \underset{\sim}{\Gamma}\, \underline{W}(k) \tag{3.12}$$

where

$$\underset{\sim}{\phi} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \tag{3.13}$$

$$\underset{\sim}{\Gamma} = \begin{bmatrix} \dfrac{T^2}{2} \\ T \end{bmatrix} \tag{3.14}$$

T is the time between measurements and equals 1 second.

The measurement equation is

$$\underline{Z}(k) = \underset{\sim}{C}\, \underline{X}(k) + \underline{V}(k)$$

$$\underline{Z}(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} \underline{X}(k) + V(k) \tag{3.15}$$

where

$X_1(k)$ is the range of the target.

$X_2(k)$ is the range rate of the target.

$V(k)$ is the measurement noise.

$w(k)$ is a random forcing input.

It is assumed that

$$E\left[v(k)\right] = 0 \tag{3.16}$$

$$E\left[v(k)^2\right] = 625 \quad (m)^2$$

$$= R \tag{3.17}$$

$$E\left[W(k)\right] = 0 \tag{3.18}$$

$$E\left[W(k)^2\right] = 225 \quad (m/sec^2)^2$$

$$= Q \tag{3.19}$$

The filter initial conditions are

$$E\left[\underline{X}(0)\right] = \begin{bmatrix} 50 \times 10^3 \ m \\ \\ -600 \ m/sec \end{bmatrix} \tag{3.20}$$

$$= \hat{\underline{X}}(0/-1) \tag{3.21}$$

And the initial covariance matrix is

$$P(0/-1) = \begin{bmatrix} 10^9 & 0 \\ \\ 0 & 10^9 \end{bmatrix} \tag{3.22}$$

Using this value for $P(0/-1)$ the gain and covariance equations were solved to determine the gain schedule $\underline{G}(k)$, $k = 0, 1, 2, \ldots,$

In the simulations three tracks (true trajectories of the target) were used with initial conditions

$$\underline{X}^{(1)}(0) = \begin{bmatrix} 60 \times 10^3 \ \text{m} \\ -600 \ \text{m/sec} \end{bmatrix} \tag{3.23}$$

$$\underline{X}^{(2)}(0) = \begin{bmatrix} 55 \times 10^3 \ \text{m} \\ -600 \ \text{m/sec} \end{bmatrix} \tag{3.24}$$

$$\underline{X}^{(3)}(0) = \begin{bmatrix} 70 \times 10^3 \ \text{m} \\ -600 \ \text{m/sec} \end{bmatrix} \tag{3.25}$$

The probabilities of occurrence of the tracks are

$$P_1 = P\left[\underline{X}(k)=\underline{X}^{(1)}(k)\right] = 0.3 \tag{3.26}$$

$$P_2 = P\left[\underline{X}(k)=\underline{X}^{(1)}(k)\right] = 0.3 \tag{3.27}$$

$$P_3 = P\left[\underline{X}(k)=\underline{X}^{(1)}(k)\right] = 0.4 \tag{3.28}$$

Figures 3.3 through 3.6 show a comparison of the results obtained by using the analytical equations and Monte-Carlo simulation. Continuous curves represent Monte-Carlo results and triangles represent the results obtained using the analytical equations. 10,000 Monte-Carlo runs were used (3000 runs for the first track, 3000 runs for the second

track and 4000 runs for the third track). From Figures 3.3 through 3.6 it is seen that the two sets of results are very close -- especially the covariances of estimation error.

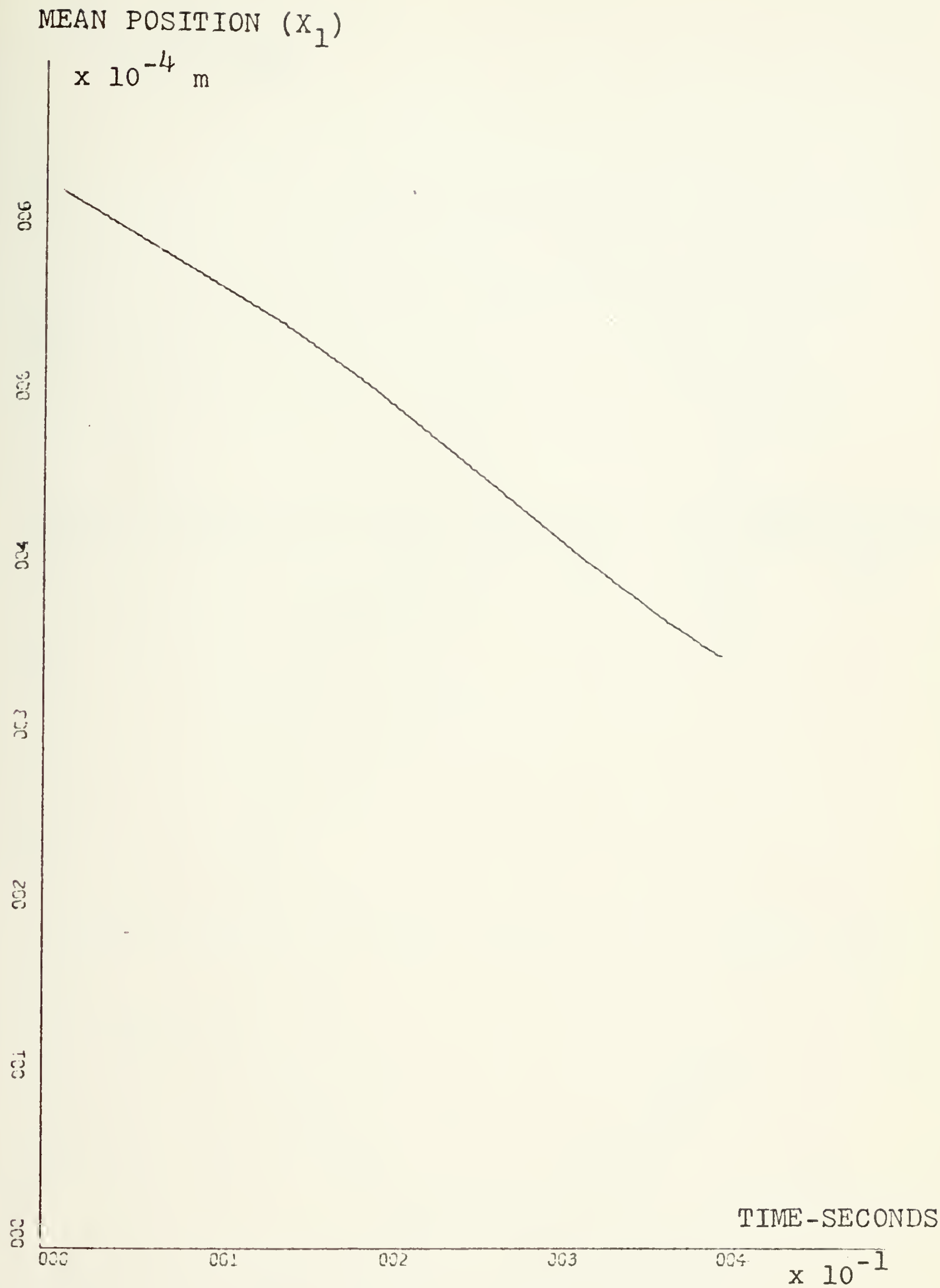

FIG. 3.1. Time history of mean position.

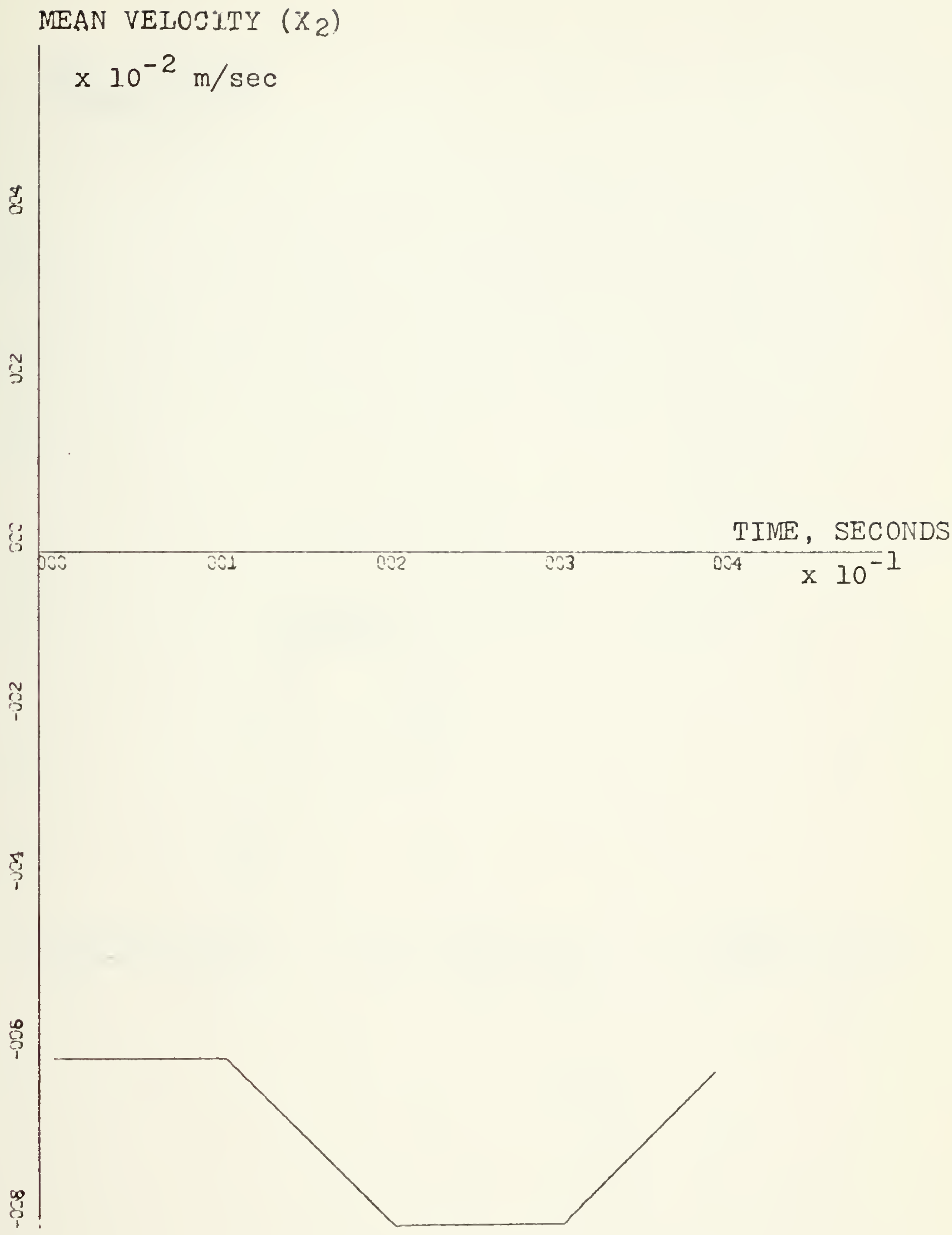

FIG. 3.2. Time history of mean velocity.

X-SCALE=1.00E+01 UNITS INCH.
Y-SCALE=2.00E+02 UNITS INCH.

FIG. 3.3. Position estimation error history.

X-SCALE=1.00E+01 UNITS INCH.
Y-SCALE=1.00E+01 UNITS INCH.

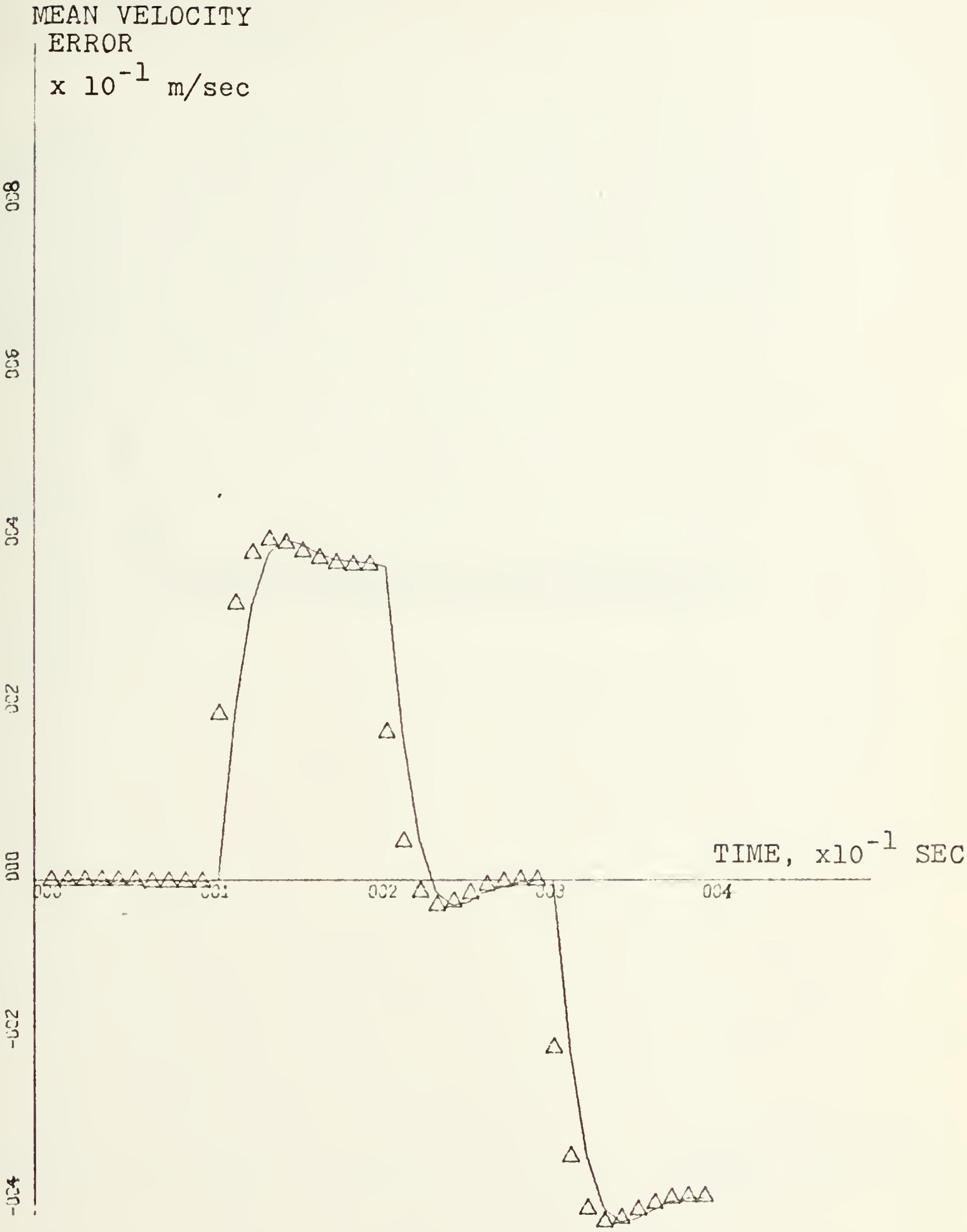

FIG. 3.4. Velocity estimation error history.

X-SCALE=1.00E+01 UNITS INCH.
Y-SCALE=2.00E+01 UNITS INCH.

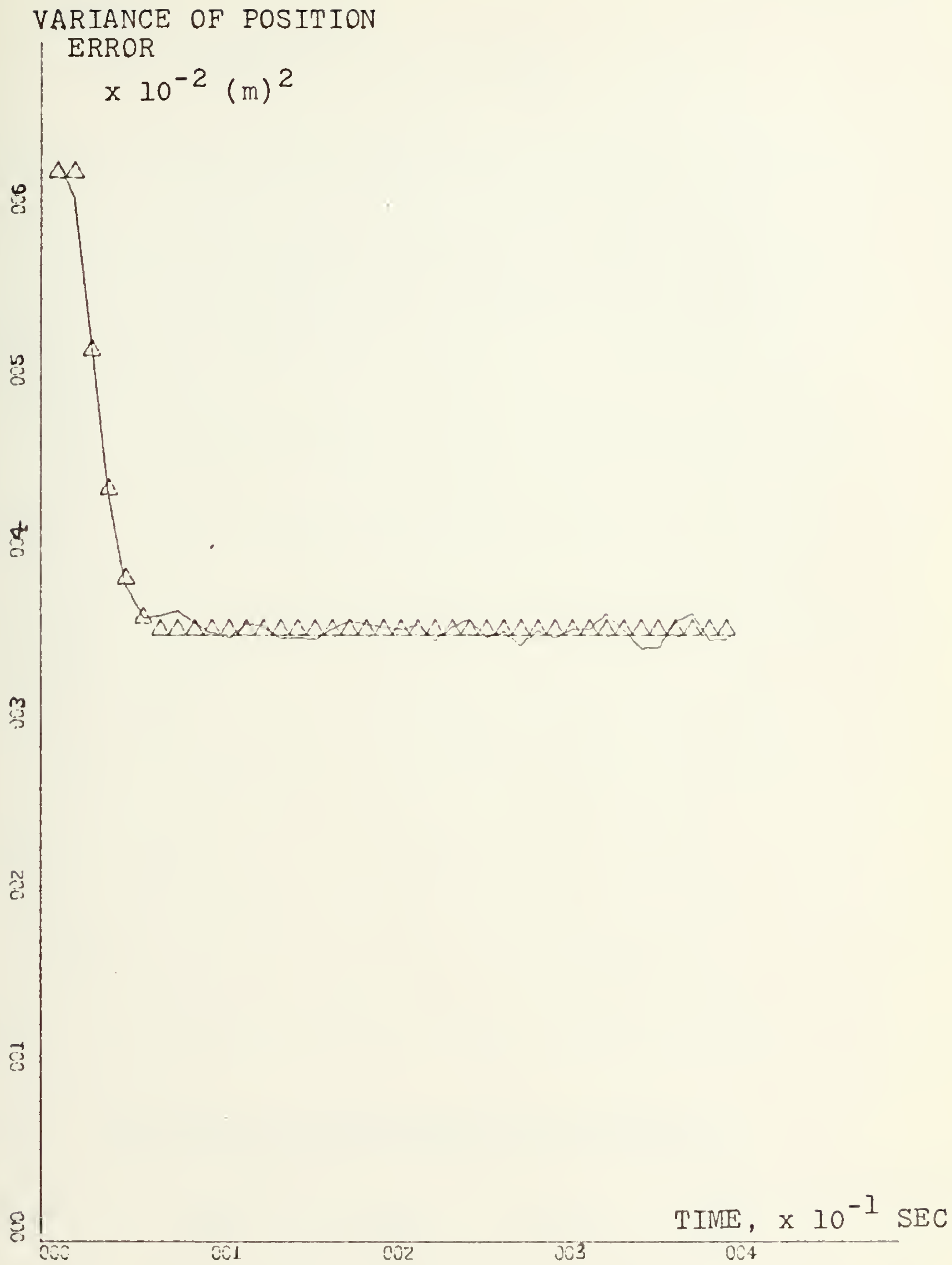

FIG. 3.5. Variance of position estimation error vs. time.

X-SCALE=1.00E+01 UNITS INCH.

Y-SCALE=1.00E+02 UNITS INCH.

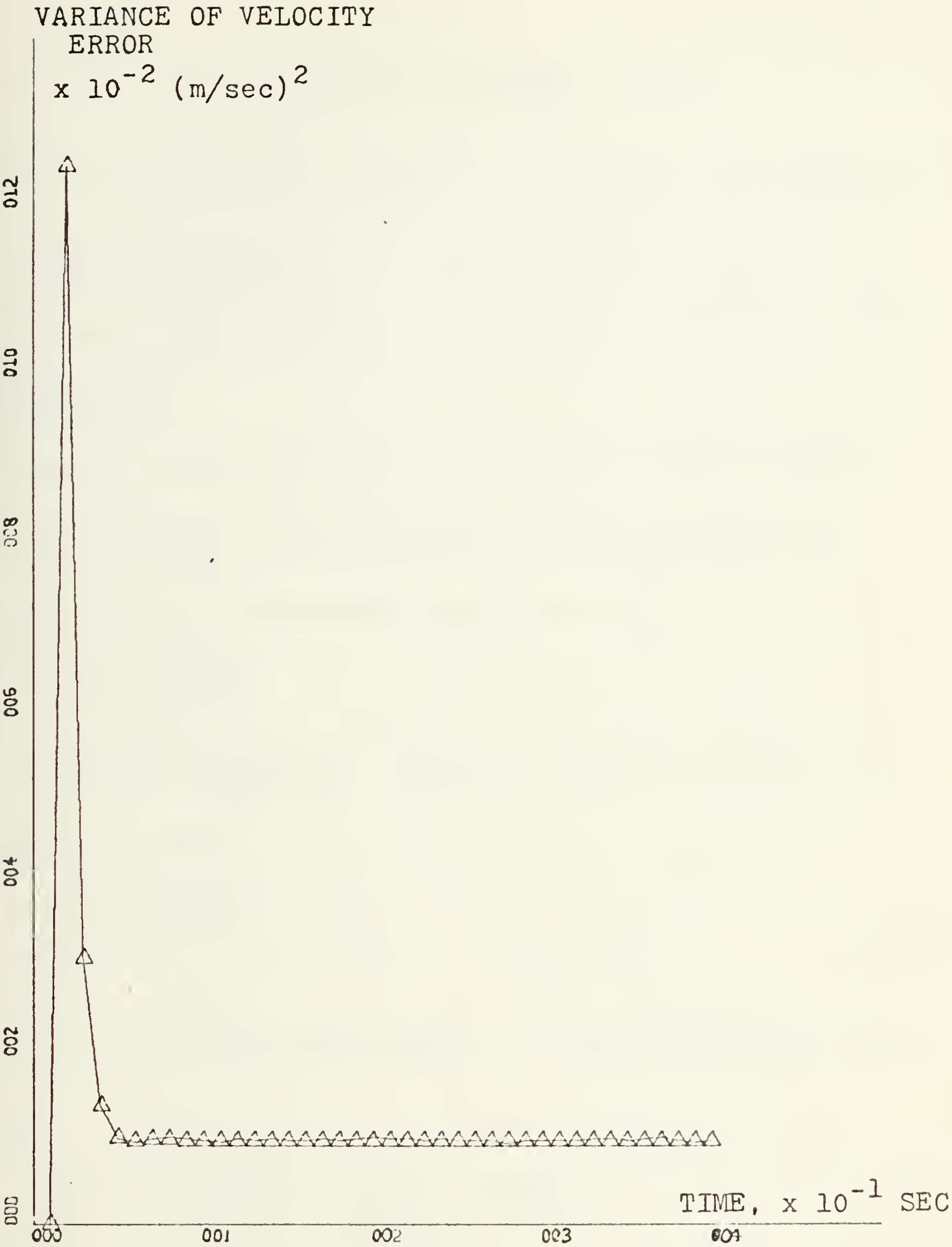

FIG. 3.6. Variance of velocity estimation error vs. time.

X-SCALE=1.00E+01 UNITS INCH.
Y-SCALE=2.00E+02 UNITS INCH.

IV. ESTIMATORS FOR NONLINEAR MODELS

The extended-Kalman filter is a commonly used approach for nonlinear estimation problems. In this chapter, the application of extended-Kalman filters to nonlinear models is summarized.

Given a continuous and nonlinear estimation problem, in order to apply linear discrete filtering theory the problem is first linearized and discretized.

A. DISCRETIZATION

Consider a nonlinear continuous system of state and observation equations given by

$$\underline{X}(t) = \underline{f}(\ \underline{X}(t),\ \underline{U}(t),\ t\ ) \tag{4.1}$$

$$\underline{Z}(t) = \underline{h}(\underline{X}(t)) + \underline{V}(t) \tag{4.2}$$

The discrete-time representation of these equations has the form

$$\underline{X}(k+1) = \underline{a}(\underline{X}(k),\ \underline{U}(k), k) + \underline{W}(k) \tag{4.3}$$

$$\underline{Z}(k) \quad = \underline{C}(\underline{X}(k)\ ) + \underline{V}(k) \tag{4.4}$$

Equation (4.3) can be obtained from (4.1) by using the relationship

$$\underline{X}(k+1) = \underline{X}(k) + (\Delta t)\cdot \underline{f}(\ \underline{X}(k),\ \underline{U}(k), k\ )$$
$$+ \frac{1}{2}\ (\Delta t)^2 \left[\frac{\partial \underline{f}}{\partial \underline{X}}\ (\underline{X}(k),\ \underline{U}(k), k) \right.$$
$$\left. \cdot \underline{f}(\underline{X}(k), \underline{U}(k), k) + \frac{\partial \underline{f}}{\partial \underline{U}}\ (\underline{X}(k),\ \underline{U}(k), k)\cdot \dot{\underline{U}}(k) \right]$$
$$+\underline{W}(k) \qquad (4.5)$$

which is the Taylor series expension of equation (4.1).

$\underline{W}(k)$ is a random input which may also be used to approximate the effect of the truncated higher-order terms.

B. LINEARIZATION

If there is available a nominal trajectory $\underline{X}(k)$ and the control $\underline{U}(k)$ is known, one can linearize equations (4.3) and (4.4) by expanding them in a taylor series about the nominal trajectory $\underline{X}'(k)$. For the state equations this gives

$$\underline{X}(k+1) = \underline{a}(\ \underline{X}'\ (k),\ \underline{U}(k), k) + \left.\frac{\partial\ \underline{a}}{\partial\ \underline{X}}\right|_{(\underline{X}'(k), \underline{U}(k), k)} \left[\underline{X}(k)\ - \underline{X}'(k)\right]$$
$$+\ \underline{W}(k) + \text{Higher order terms (H.O.T)} \qquad (4.6)$$

Defining the matrix $\underset{\sim}{\emptyset}(k)$ as

$$\underset{\sim}{\emptyset}(k) = \left.\frac{\partial\ \underline{a}}{\partial\ \underline{X}}\right|_{(\underline{X}'(k), \underline{U}(k), K)} \qquad (4.7)$$

and truncating the higher-order terms, Equation (4.6) reduces to

$$\underline{X}(k+1) = \underset{\sim}{\emptyset}(k)\ \underline{X}(k) + \underline{a}(\underline{X}'(k), \underline{U}(k), k)$$
$$-\ \underset{\sim}{\emptyset}(k)\ \underline{X}'(k) + \underline{W}(k) \qquad (4.8)$$

Since $\underline{X}'(k)$ is a known quantity, the second and third terms of the right hand side of equation (4.8) are known and deterministic. Defining the vector $\underline{U}'(k)$ as

$$\underline{U}'(k) = \underline{a}(\underline{X}'(k),\underline{U}(k),k) - \emptyset(k)\ \underline{X}'(k) \tag{4.9}$$

Equation (4.8) becomes

$$\underline{X}(k+1) = \underset{\sim}{\emptyset}(k)\ \underline{X}(k) + \underline{U}'(k) + \underline{W}(k) \tag{4.10}$$

Applying the same procedure to Equation (4.4), the linearized form of the observation equation is

$$\underline{Z}(k) = \underset{\sim}{H}(k)\ \underline{X}(k) + \underline{C}(\underline{X}'(k)\ ) - \underset{\sim}{H}(k)\ \underline{X}'(k) + \underline{V}(k) \tag{4.11}$$

where

$$\underline{H}(k) = \left. \frac{\partial \underline{C}}{\partial \underline{X}} \right|_{\underline{X}'(k)} \tag{4.12}$$

Defining the vector $\underline{B}(k)$ as

$$\underline{B}(k) = \underline{C}(\underline{X}'(k)) - \underset{\sim}{H}(k)\ \underline{X}'(k) \tag{4.13}$$

yields

$$\underline{Z}(k) = \underset{\sim}{H}(k)\ \underline{X}(k) + \underline{B}(k) + \underline{V}(k) \tag{4.14}$$

Equations (4.10) and (4.14) represent a linear time-varying model, $\underline{U}'(k)$ and $\underline{B}(k)$ represent bias terms resulting from the linearization process. The analytical equations can be applied to this model by replacing the $\underline{U}(k-1)$ term in (2.15) with the $\underline{U}'(k-1)$ term in Equation (2.37) and defining new measurement equations given by

$$\underline{Z}'(k) = \underline{Z}(k) - \underline{B}(k)$$

$$= \underset{\sim}{H}(k)\ \underline{X}(k) + \underline{V}(k) \qquad (4.15)$$

C. THE EXTENDED-KALMAN FILTER

Consider a nonlinear discrete system of state and observation equations given by

$$\underline{X}(k+1) = \underline{f}(\underline{X}(k), k\ ) + \underline{W}(k) \qquad (4.16)$$

$$\underline{Z}(k) = \underline{C}(\ \underline{X}(k)\ ) + \underline{V}(k) \qquad (4.17)$$

In these equations $\underline{f}$ and $\underline{C}$ are nonlinear functions of the state variables $\underline{X}(k)$, $\underline{W}(k)$ is a random forcing input and $\underline{V}(k)$ is measurement noise with the usual assumptions (an uncorrelated, zero mean random processes).

$$E\left[\underline{W}(k)\ \underline{W}(j)^T\right] = Q(k)\ \delta_{kj}$$

$$E\left[\underline{V}(k)\ \underline{V}(j)^T\right] = R(k)\ \delta_{kj}$$

In order to apply the linear filter equations, Equations (4.16) and (4.17) are expanded about the nominal trajectory $\underline{X}'(k)$ if it is available. In practice, it is possible to have an idea about the target trajectory for some problems, like satellite orbit determination problems, in which case one may predict the target trajectory to be very close to the true trajectory and thus, be able to define a nominal trajectory for the purpose of linearization. In such cases the gain schedule can be computed before estimation, but in

other kinds of problems, it may be impossible to obtain a nominal trajectory. An alternative is to linearize the problem at each time point about the best estimates of the states currently available. In this case the gains can only be calculated for each sample as the estimates are available. This approach is called the extended-Kalman filter.

Using an estimate to evaluate the linearization Equation (4.16) gives

$$\underline{X}(k+1) = \underset{\sim}{\phi}(k)\ \underline{X}(k) + \underline{W}(k) \tag{4.18}$$

where

$$\underset{\sim}{\phi}(k) = \left.\frac{\partial \underline{f}}{\partial \underline{X}}\right|_{\hat{\underline{X}}(k/k)} \tag{4.19}$$

Similarly Equation (4.17) yields

$$\underline{Z}(k) = \underset{\sim}{H}(k)\ \underline{X}(k) + \underline{V}(k) \tag{4.20}$$

where

$$\underset{\sim}{H}(k) = \left.\frac{\partial \underline{C}}{\partial \underline{X}}\right|_{\hat{\underline{X}}(k/k-1)} \tag{4.21}$$

$\underset{\sim}{\phi}(k)$ and $\underset{\sim}{H}(k)$ are obtained as indicated in the linearization process discussed earlier and evaluated using the estimate as the nominal track. The filter estimation equation is

$$\hat{\underline{X}}(k/k) = \hat{\underline{X}}(k/k-1) + \underset{\sim}{G}(k) \left[\ \underline{Z}(k) - \underset{\sim}{C}(\underline{X}(k/k-1))\right] \tag{4.22}$$

The prediction equation is

$$\hat{\underline{X}}(k/k-1) = \underline{f}(\ \hat{\underline{X}}(k-1/k-1),\ k-1) \tag{4.23}$$

The gains are obtained by using the relationship

$$\underset{\sim}{G}(k) = \underset{\sim}{P}(k/k-1)\ \underset{\sim}{H}(k)^T \left[\underset{\sim}{H}(k)\ \underset{\sim}{P}(k/k-1)\ \underset{\sim}{H}(k)^T + \underset{\sim}{R}(k) \right]^{-1} \tag{4.24}$$

where

the theoretical covariance equation is

$$\underset{\sim}{P}(k/k) = \left[\underset{\sim}{I} - \underset{\sim}{G}(k)\ \underset{\sim}{H}(k) \right] \underset{\sim}{P}(k/k-1) \tag{4.25}$$

The covariance propagation equation is

$$\underset{\sim}{P}(k/k-1) = \underset{\sim}{\phi}(k-1)\ \underset{\sim}{P}(k-1/k-1)\ \underset{\sim}{\phi}(k-1)^T + \underset{\sim}{Q}(k-1) \tag{4.26}$$

and the initial conditions are

$$\begin{aligned} \hat{\underline{X}}(0/-1) &= E\left[\underline{X}(0)\right] \\ &= \bar{\underline{X}}_0 \\ \underset{\sim}{P}(0/-1) &= \underset{\sim}{P}_0 \\ &= E\left\{ \left[\underline{X}(0) - \bar{\underline{X}}_0 \right] \left[\underline{X}(0) - \bar{\underline{X}}_0 \right]^T \right\} \end{aligned} \tag{4.27}$$

V. APPLICATION OF THE ANALYTICAL EQUATIONS TO EXTENDED-KALMAN FILTERS: A RE-ENTRY PROBLEM

In this chapter, the approximate evaluation of the performance of an extended-Kalman filter using the analytical equations is discussed. Since the analytical equations are based on linear estimators with pre-computed gain schedules, application of these equations to a problem with gains which are not pre-computed is an approximation.

In an extended-Kalman filter the gains are evaluated on-line using the estimates as they are produced. This means that the gains will vary from run-to-run even if the track is the same. This variation is caused by the differences in measurement noise which in turn effect the estimates. To use the analytical equation approach to evaluate an extended-Kalman filter it is assumed that the linearization which results in Equations (4.10) and (4.15) is performed using the true track. This leads to a pre-computed gain schedule that is used to approximate the extended-Kalman filter gain schedule.

A. THE RE-ENTRY PROBLEM

In order to compare the results of using the analytical equations and Monte-Carlo simulation, a particular problem was selected which contains significant nonlinearities in both the state and observation equations. This problem deals with estimation of the altitude, velocity and constant

ballistic coefficient of a vertically falling body. The measurements are taken at discrete time intervals of 1 second by a radar that measures range in the presence of (discrete) white gaussian noise. The geometry of the problem is illustrated in Figure 5.1. It is assumed that the body is falling vertically.

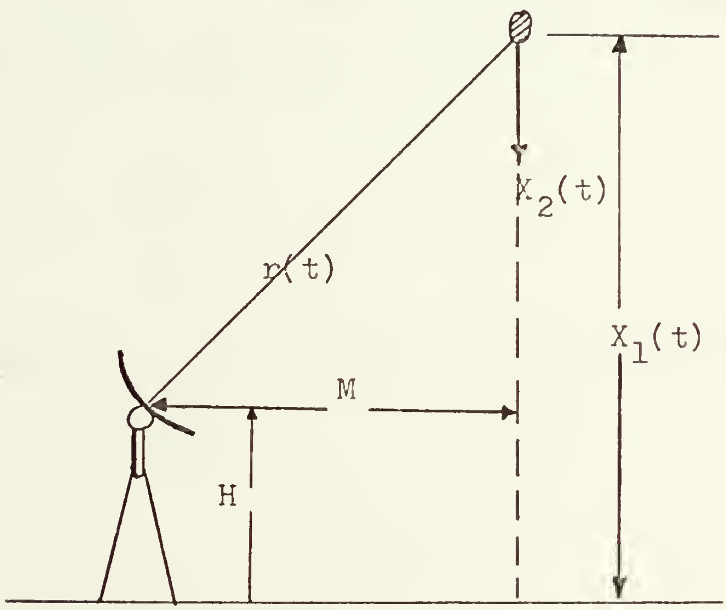

FIG. 5.1. Geometry of Re-entry problem.

The following definitions are used:

$X_1(t)$	:	Altitude
$X_2(t)$	:	Velocity (downward)
m	:	Mass (constant)
C_D	:	Drag coefficient (constant)
A	:	Reference area for drag evaluation (constant)
ρ	:	Mass density of the atmosphere
H	:	Radar altitude
M	:	Horizontal distance
$r(t)$	:	True range

It is assumed that the effect of gravity is negligible.

The equations of motion are

$$\dot{X}_1(t) = -X_2(t)$$

$$\dot{X}_2(t) = -\frac{C_D A \rho}{2m} X_2(t)^2 \qquad (5.1)$$

The air density is approximated by the exponential function

$$\rho = \rho_o e^{-\gamma X_1(t)} \qquad (5.2)$$

where

$$\gamma = 5 \times 10^{-5} \qquad (5.3)$$

Defining

$$X_3(t) = \frac{C_D A \rho_o}{2m} \qquad (5.4)$$

which is a constant ballistic parameter, the state equations become

$$\dot{X}_1(t) = -X_2(t)$$

$$\dot{X}_2(t) = -\,X_2(t)^2\,X_3(t)\,e^{-\gamma X_1(t)} \tag{5.5}$$

$$\dot{X}_3(t) = 0$$

which are of the form

$$\dot{\underline{X}}(t) = \underline{f}(\,\underline{X}(t)\,) \tag{5.6}$$

The measurement r(t) (range) is given by

$$r(t) = \sqrt{M^2 + (X_1(t) - H\,)^2} \tag{5.7}$$

and is observed at discrete instants of time so that the observed sequence is

$$z(k) = \sqrt{M^2 + (X_1(k) - H\,)^2} + v(k) \tag{5.8}$$

where v(k) is white gaussian noise with zero mean and constant variance

$$E\left[\,v(k)^2\right] = R$$

In this case, the output nonlinearity has the form

$$C(\,\underline{X}(k)\,) = \sqrt{M^2 + (X_1(k) - H\,)^2} \tag{5.9}$$

It is assumed that

$$H = 0$$

$$M = 100{,}000 \text{ ft}$$

and the filter initial conditions are

$$\hat{\underline{X}}(0/-1) = \overline{\underline{X}}_o$$

$$= \begin{bmatrix} 3 \times 10^5 \text{ ft} \\ 2 \times 10^4 \text{ ft/sec} \\ 3 \times 10^{-5} \text{ ft}^{-1} \end{bmatrix} \tag{5.10}$$

The assumed initial covariance matrix is

$$\underset{\sim}{P}(0/-1) = \begin{bmatrix} 10^6 & 0 & 0 \\ 0 & 4\text{x}10^6 & 0 \\ 0 & 0 & 10^{-4} \end{bmatrix} \tag{5.11}$$

and the variance of measurement noise is

$$R = 10 \times 10^4 \quad (\text{ft})^2 \tag{5.12}$$

The true initial conditions of the falling body are

$$\underline{X}(0) = \begin{bmatrix} 3\text{x}10^5 & \text{ft} \\ 2\text{x}10^4 & \text{ft/sec} \\ 1\text{x}10^{-3} & \text{ft}^{-1} \end{bmatrix}$$

Applying Equation (4.5) to Equation (5.5) it can be shown that the discretized state equations are

$$\underline{X}(k+1) = \underline{a}(\ \underline{X}(k)\ ) \tag{5.14}$$

$$X_1(k+1) = X_1(k) - X_2(k) + \tfrac{1}{2} X_2\ (k)^2\ X_3(k)\ e^{-\gamma X_1(k)} \tag{5.15}$$

$$X_2(k+1) = X_2(k) - X_2(k)^2\ X_3(k)\ e^{-\gamma X_1(k)} - \tfrac{1}{2}\ X_2(k)^3\ X_3(k) \cdot e^{-\gamma X_1(k)} + X_2(k)^3\ X_3(k)^2\ e^{-2\gamma X_1(k)} \tag{5.16}$$

$$X_3(k+1) = X_3(k) \tag{5.17}$$

Linearizing Equations (5.15) - (5.17) it can be shown that the $\emptyset(k)$ matrix is

$$\emptyset(k) = \left.\frac{\partial \underline{a}}{\partial \underline{X}}\right|_{\underline{X}'(k)}$$

$$= \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \tag{5.18}$$

where $\underline{X}'(k)$ is the nominal trajectory and the A_{ij}'s are

$$A_{11} = 1 - (1/2)\ X_2'(k)^2\ X_3'(k)\ e^{-\gamma X_1'(k)} \tag{5.19}$$

$$A_{12} = -1 + X_2'(k)\ X_3'(k)\ e^{-\gamma X_1'(k)} \tag{5.20}$$

$$A_{13} = \frac{1}{2} X_2'(k)^2\ e^{-\gamma X_1'(k)} \tag{5.21}$$

$$A_{21} = X_2'(k)^2\ X_3'(k)\ e^{-\gamma X_1'(k)} + \frac{1}{2}\gamma^2\ X_2'(k)^3\ X_3'(k) e^{-\gamma X_1'(k)}$$

$$- 2\gamma X_2'(k)^3\ X_3'(k)^2\ e^{-2\gamma X_1'(k)} \tag{5.22}$$

$$A_{22} = 1 - 2\ X_2'(k)\ X_3'(k)\ e^{-\gamma X_1'(k)}$$

$$- \frac{3}{2}\gamma\ X_2'(k)^2\ X_3'(k)\ e^{-\gamma X_1'(k)}$$

$$+ 3\ X_2'(k)^2\ X_3'(k)^2\ e^{-2\gamma X_1'(k)} \tag{5.23}$$

$$A_{23} = -X_2'(k)^2 \, e^{-\gamma X_1'(k)} - \frac{1}{2} \gamma \, X_2'(k)^3 \, e^{-\gamma X_1'(k)}$$

$$+ 2 \, X_2'(k)^3 \, X_3'(k) \, e^{-2\gamma X_1'(k)} \tag{5.24}$$

$$A_{31} = 0 \tag{5.25}$$

$$A_{32} = 0 \tag{5.26}$$

$$A_{33} = 1 \tag{5.27}$$

The $\underset{\sim}{H}(k)$ matrix is

$$\underset{\sim}{H}(k) = \left. \frac{\partial \underline{C}}{\partial \underline{X}} \right|_{\underline{X}'(k)}$$

$$= \begin{bmatrix} \dfrac{X_1'(k) - H}{\sqrt{M^2 + (X_1'(k) - H)^2}} & 0 & 0 \end{bmatrix} \tag{5.28}$$

In the simulation of the problem, the true track was generated by using the discretized state equations (5.15) through (5.17). The linearization is made about the track in the application of the analytical equations and $\underline{U}'(k)$ is defined as

$$\underline{U}'(k) = \underline{a}(\, \underline{X}'(k), k \,) - \underset{\sim}{\phi}(k) \, \underline{X}'(k) \tag{5.29}$$

If the true track is generated by solving the state equations then

$$\underline{X}'(k+1) = \underline{a}(\ \underline{X}'(k),k\ ) \tag{5.30}$$

and

$$\underline{U}'(k) = \underline{X}'(k+1) - \underset{\sim}{\phi}(k)\ \underline{X}'(k) \tag{5.31}$$

$$\underline{U}'(k-1) = \underline{X}'(k) - \underset{\sim}{\phi}(k-1)\ \underline{X}'(k-1) \tag{5.32}$$

If one uses a track other than that generated by solving the state equations, then Equation (5.29) must be used for $\underline{U}'(k)$. The matrices $\underset{\sim}{\phi}(k)$, $\underset{\sim}{H}(k)$ and the vector $\underline{U}'(k-1)$ can be used for application of the analytical equations. In the Monte-Carlo simulation of the extended-Kalman filter, one must replace $\underline{X}'(k)$ with the estimates $\hat{\underline{X}}(k/k)$ in Equations (5.19) through (5.27) and by $\hat{\underline{X}}(k/k-1)$ in Equation (5.28) ($\underset{\sim}{\phi}(k)$ must be evaluated about $\hat{\underline{X}}(k/k)$ and $\underset{\sim}{H}(k)$ must be evaluated about $\hat{\underline{X}}(k/k-1)$).

Figures 5.5 through 5.10 illustrate the results of the analytical equations and the Monte-Carlo simulation of the extended-Kalman filter. The continuous curves represent the Monte-Carlo results (1000 runs) and the triangles represent results from the analytical equations. The figures show that the analytical equations have predicted better performance than that predicted by the Monte-Carlo simulation. Actually, this observation is not generally true. The difference between results of the two methods

FIG. 5.2. Altitude history of re-entry vehicle.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=5.00E+04 UNITS INCH.

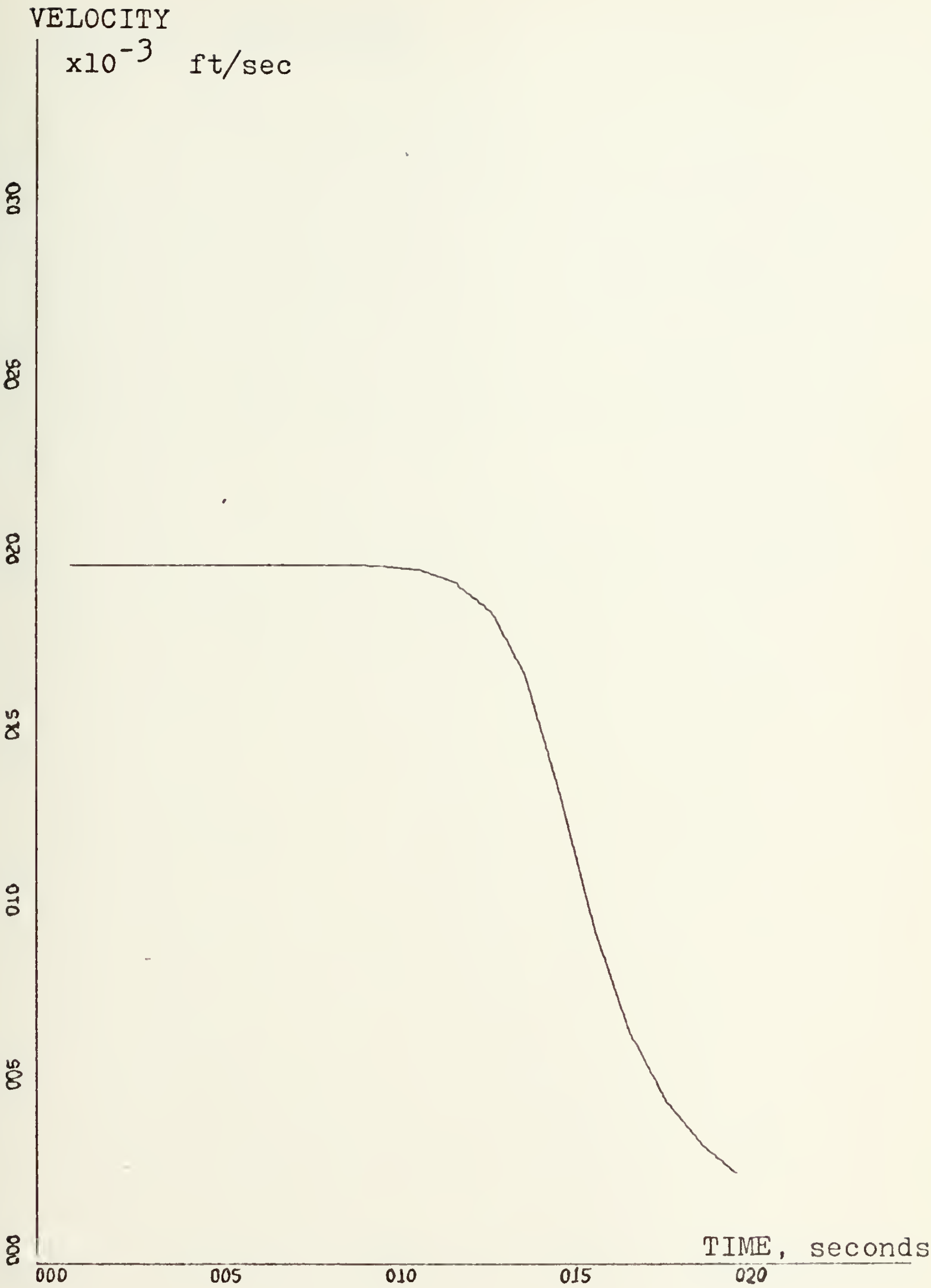

FIG. 5.3. Velocity history of re-entry vehicle.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=5.00E+03 UNITS INCH.

FIG. 5.4. Time history of ballistic coefficient.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=2.00E-04 UNITS INCH.

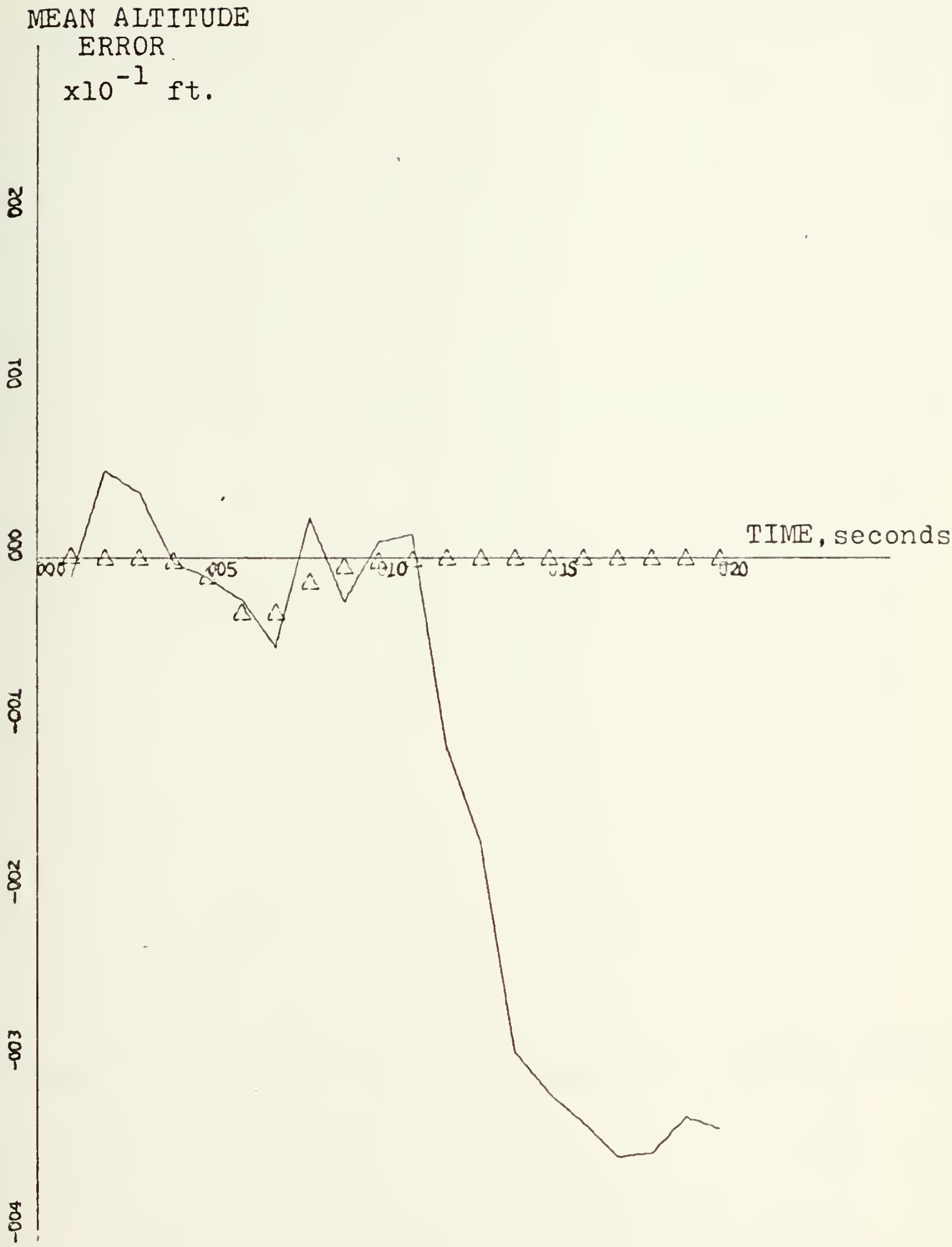

FIG. 5.5. Time history of altitude estimation error.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=1.00E+01 UNITS INCH.

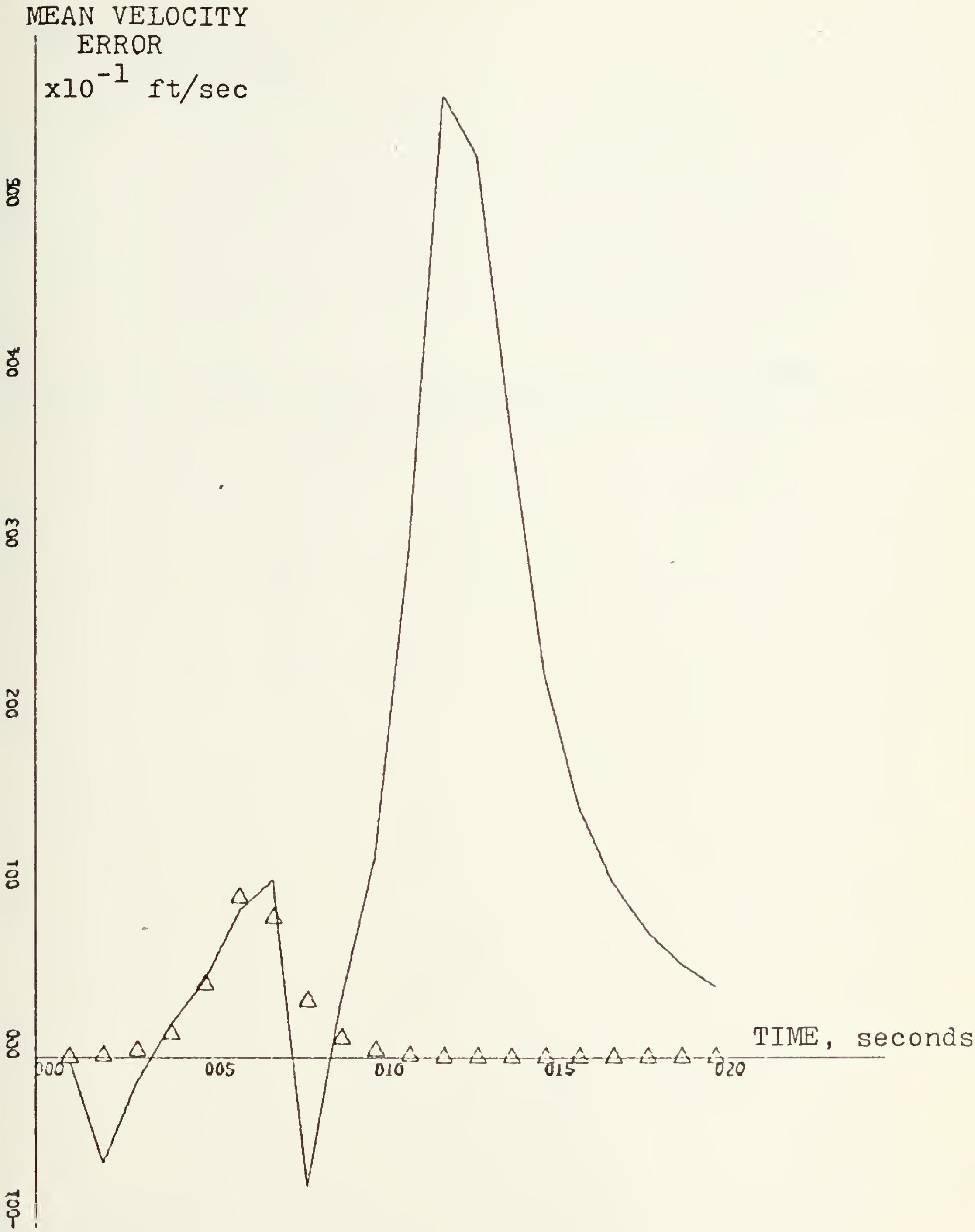

FIG. 5.6. Time history of the velocity estimation error.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=1.00E+01 UNITS INCH.

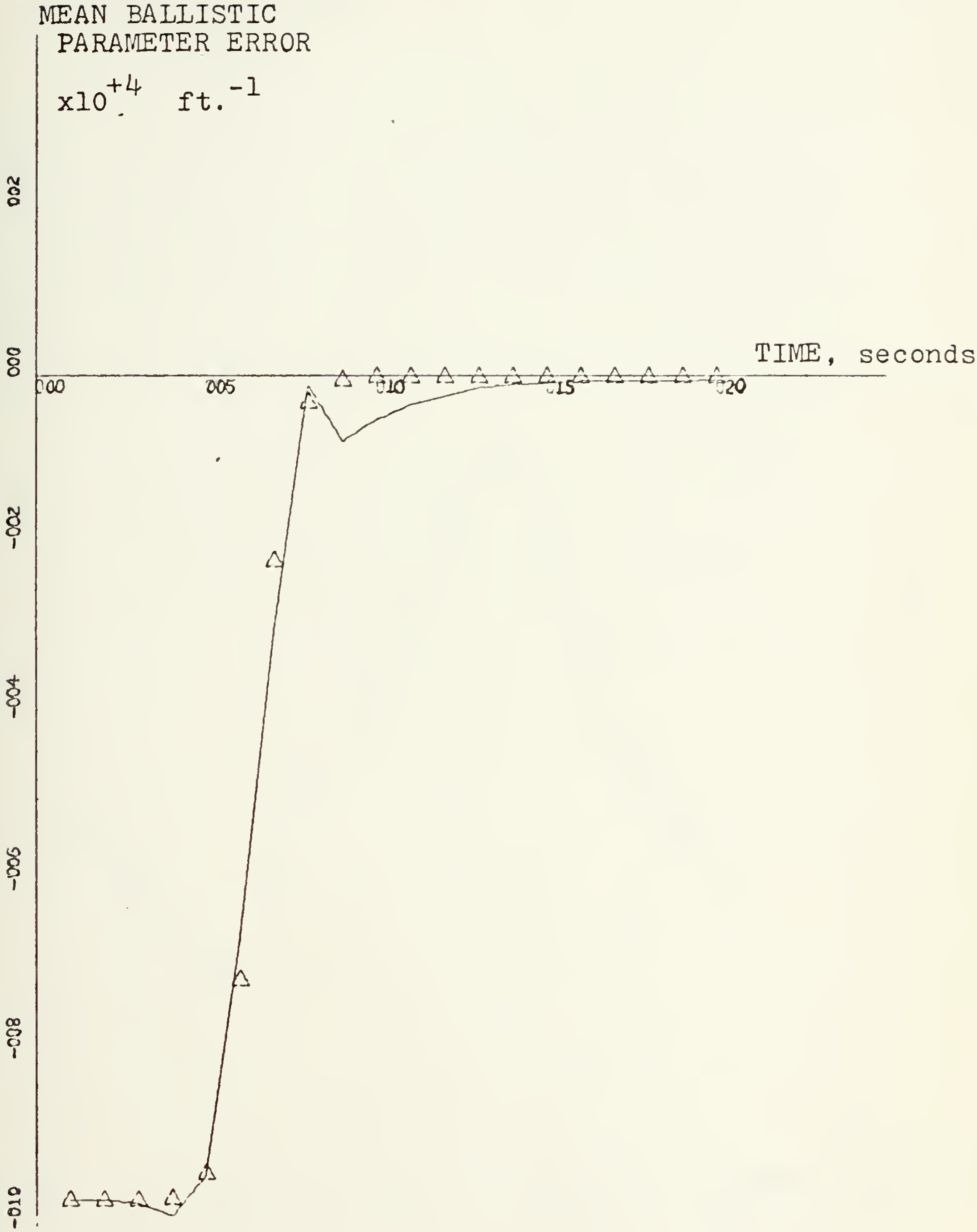

FIG. 5.7. Time history of the mean ballistic parameter estimation error.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=2.00E-04 UNITS INCH.

FIG. 5.8. Variance of altitude estimation error vs. time.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=5.00E+03 UNITS INCH.

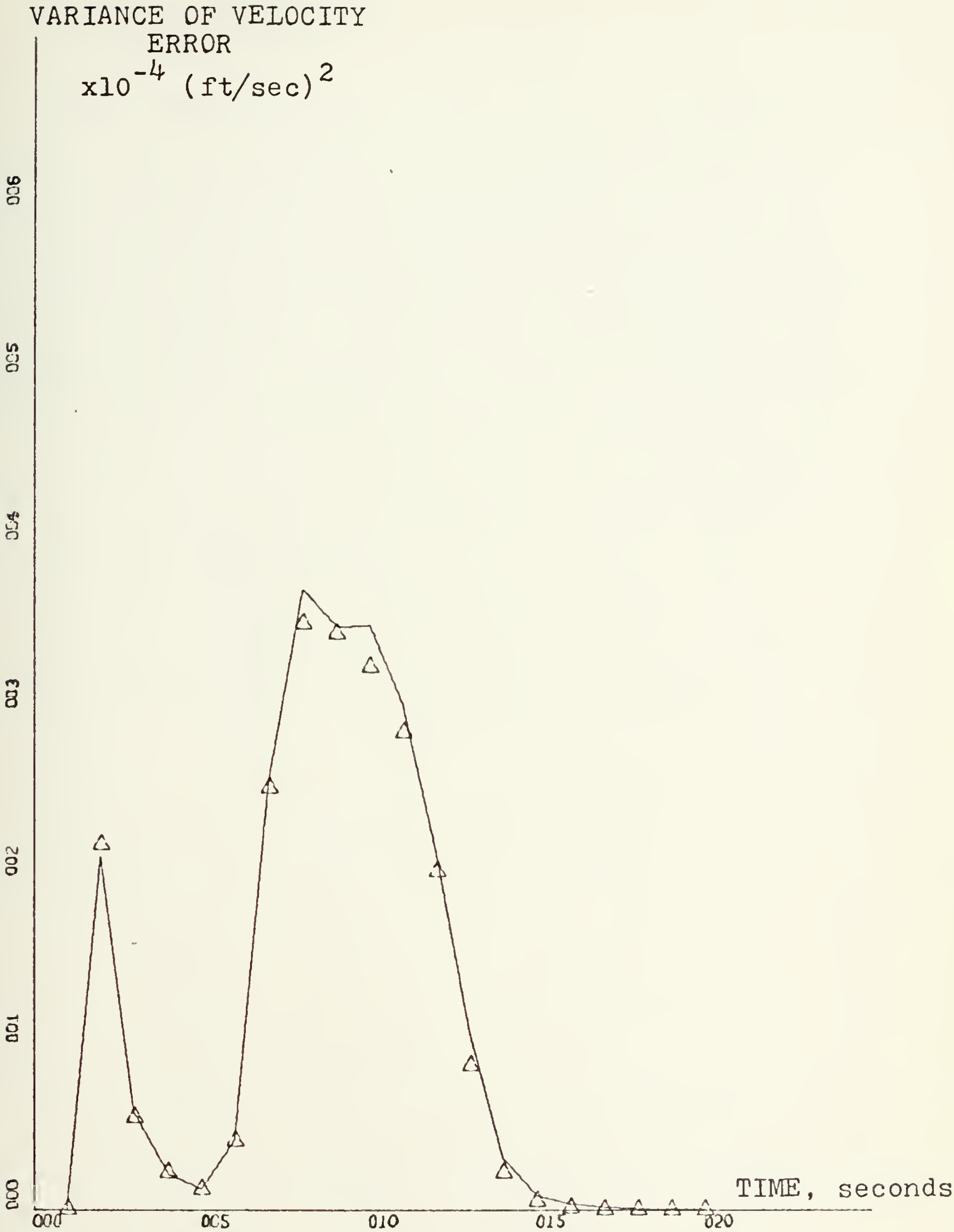

FIG. 5.9. Variance of velocity estimation error vs. time.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=1.00E+04 UNITS INCH.

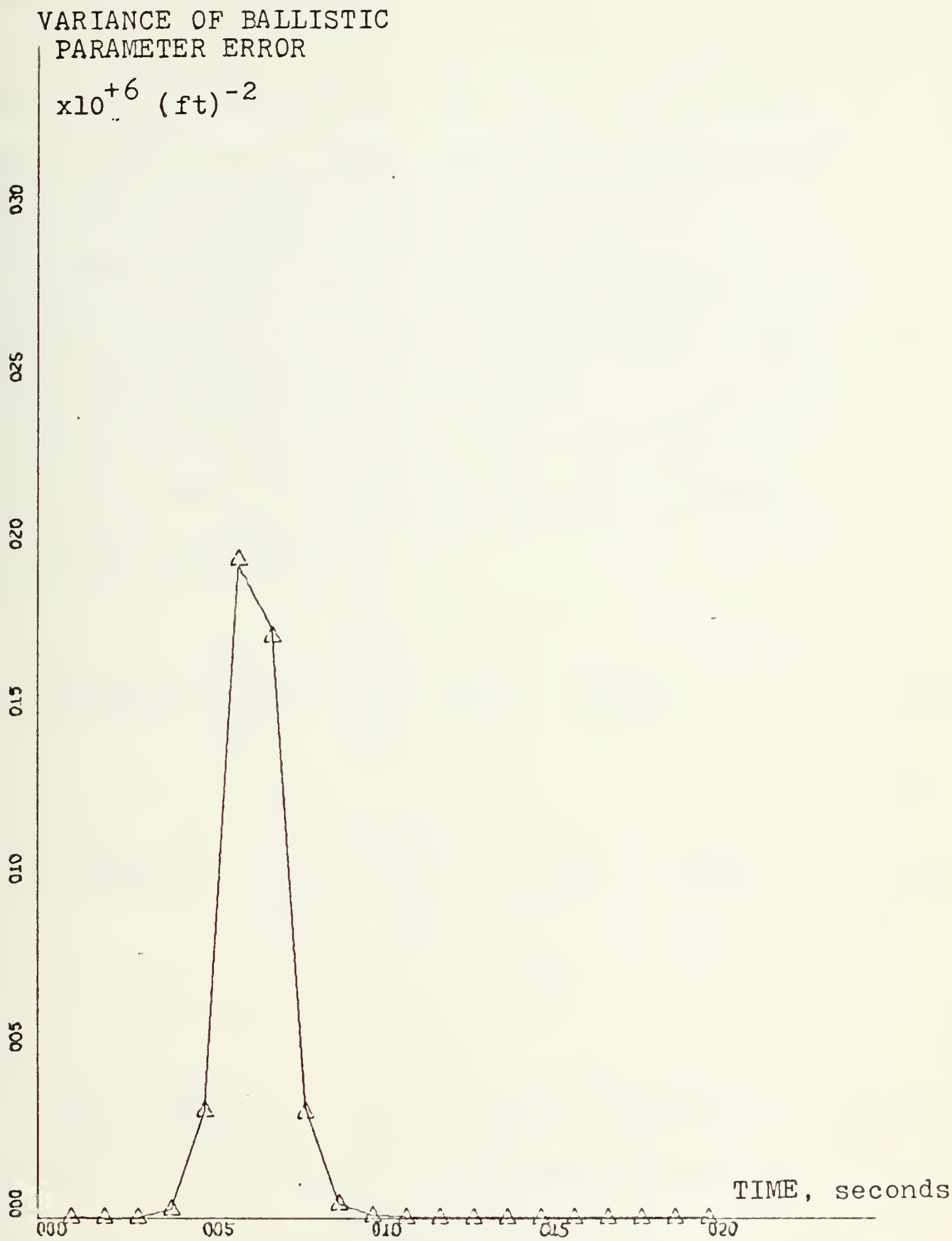

FIG. 5.10. Variance of ballistic parameter estimation error vs. time.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=5.00E-06 UNITS INCH.

depends on the measurement noise, the nature of the nonlinearities and the filter initial conditions. For the re-entry problem the two results are very close. For example, at 50,000 ft altitude, the mean altitude error difference between the two results is 30 feet.

In order to investigate the noise dependence of the difference between two results, the problem has been solved by each method for various covariances of the measurement noise and the root mean square of the difference has been calculated. The root mean square of the difference between two results is defined as

$$\Delta\tilde{\underline{\mu}}_{rms}(I) = \left\{ \frac{1}{N} \sum_{k=1}^{N} \left[\tilde{\underline{\mu}}_1(k) - \tilde{\underline{\mu}}_2(k) \right]^2 \right\}^{1/2} \tag{5.33}$$

$$\Delta \underline{VAR}_{rms}(I) = \left\{ \frac{1}{N} \sum_{k=1}^{N} \left[\underline{VAR}_1(k) - \underline{VAR}_2(k) \right]^2 \right\}^{1/2} \tag{5.34}$$

where

$\tilde{\underline{\mu}}_1(k)$, $\underline{VAR}_1(k)$ are the mean and variance of estimation error calculated by the Monte-Carlo simulation,

$\tilde{\underline{\mu}}_2(k)$, $\underline{VAR}_2(k)$ are the mean and variance of estimation error calculated by the analytical equations,

N is the number of time points, and

$\Delta\tilde{\mu}_{rms}(I), \Delta \underline{VAR}_{rms}(I)$ are the rms values of the difference between the two results.

The results obtained have been tabulated in Table 5.1. From, Table 5.1 it is seen that the root mean square differences between the two results for both altitude and velocity estimation error increase as the covariance of measurement noise increases. But this is not true for X_3 which is the ballistic coefficient. Table 5.1 indicates that estimation of the ballistic coefficient is relatively insensitive to the measurement noise level.

Figures 5.11 through 5.16 represent the results of 50 Monte-Carlo runs (continuous curves) and the analytical equations. It is seen that there is significant difference between 1000 Monte-Carlo run-results and 50 Monte-Carlo run-results. (See Figure 5.5-5.10).

Variance of measurement noise	$\Delta\mu_{rms}(1)$ (feet)	$\Delta\mu_{rms}(2)$ (feet/sec)	$\Delta\mu_{rms}(3)$ (1/feet)	$\Delta VAR_{rms}(1)$ $(feet)^2$	$\Delta VAR_{rms}(2)$ $(feet/sec)^2$	$\Delta VAR_{rms}(3)$ $(1/feet)^2$
1	8.15043×10^{-2}	4.1566×10^{-2}	4.04867×10^{-5}	5.2617×10^{-2}	0.11749	3.0333×10^{-8}
100	0.34382	0.5678	4.2500×10^{-5}	4.8689	7.2385	1.2267×10^{-7}
1000	2.5657	3.5499	2.1497×10^{-5}	47.8941	73.7134	5.0683×10^{-8}
10,000	20.021	21.7225	3.1718×10^{-5}	616.742	989.287	7.0329×10^{-8}

TABLE 5.1. RMS deviations of means and variances obtained by Monte-Carlo and Analytical methods for various variances of measurement noise.

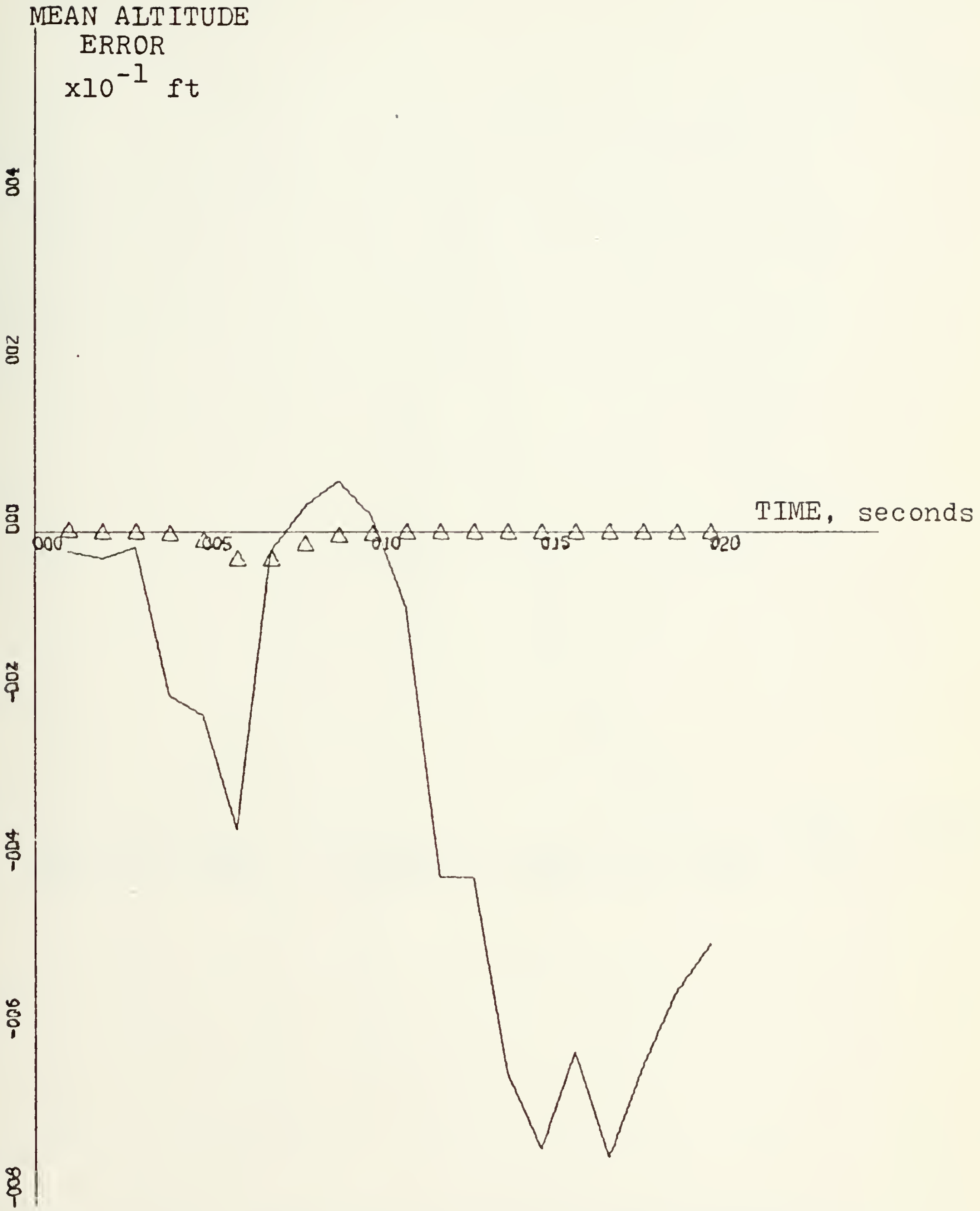

FIG. 5.11. Time history of the mean altitude estimation error for 50 Monte-Carlo runs.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=2.00E+01 UNITS INCH.

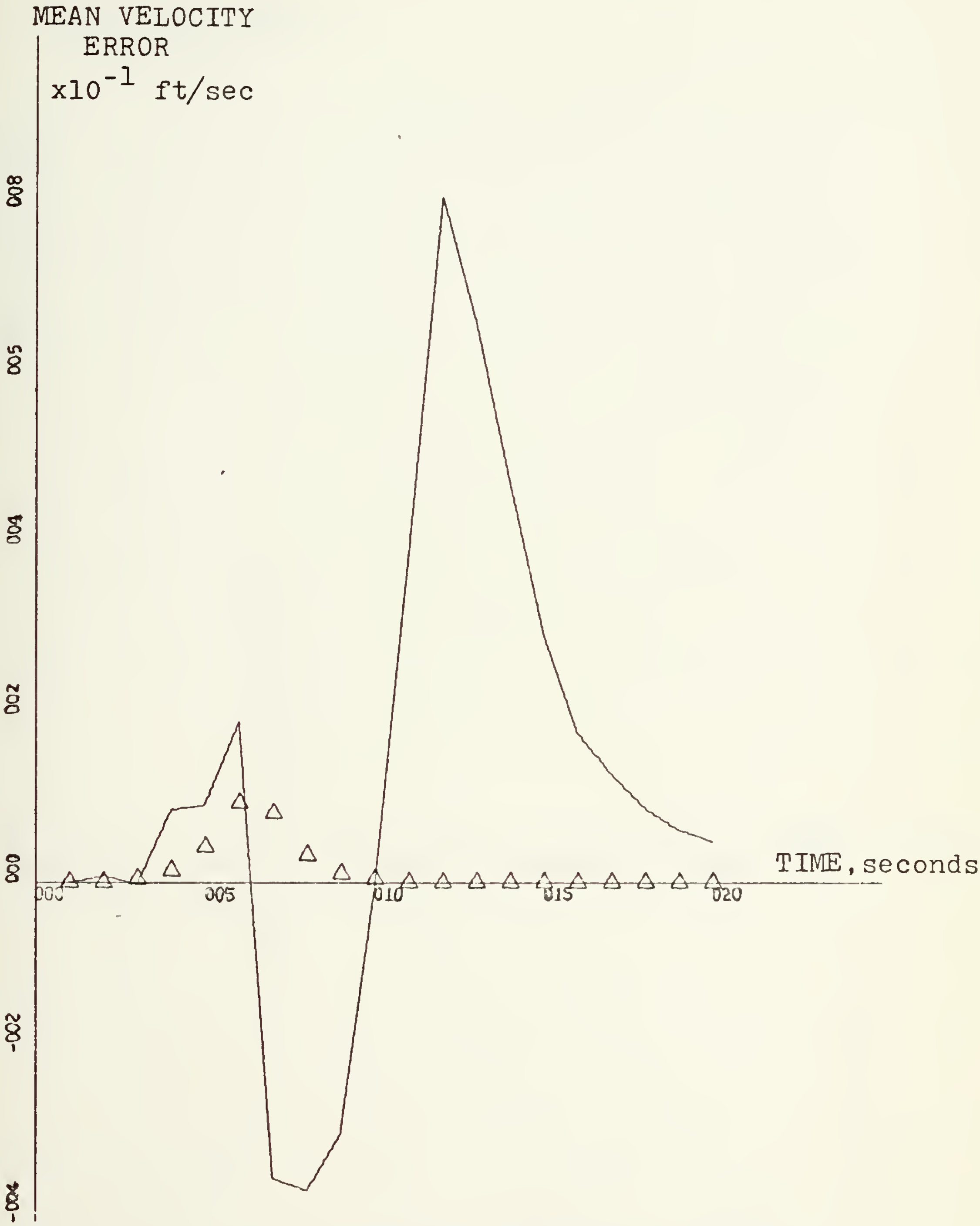

FIG. 5.12. Time history of the mean velocity estimation error for 50 Monte-Carlo runs.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=2.00E+01 UNITS INCH.

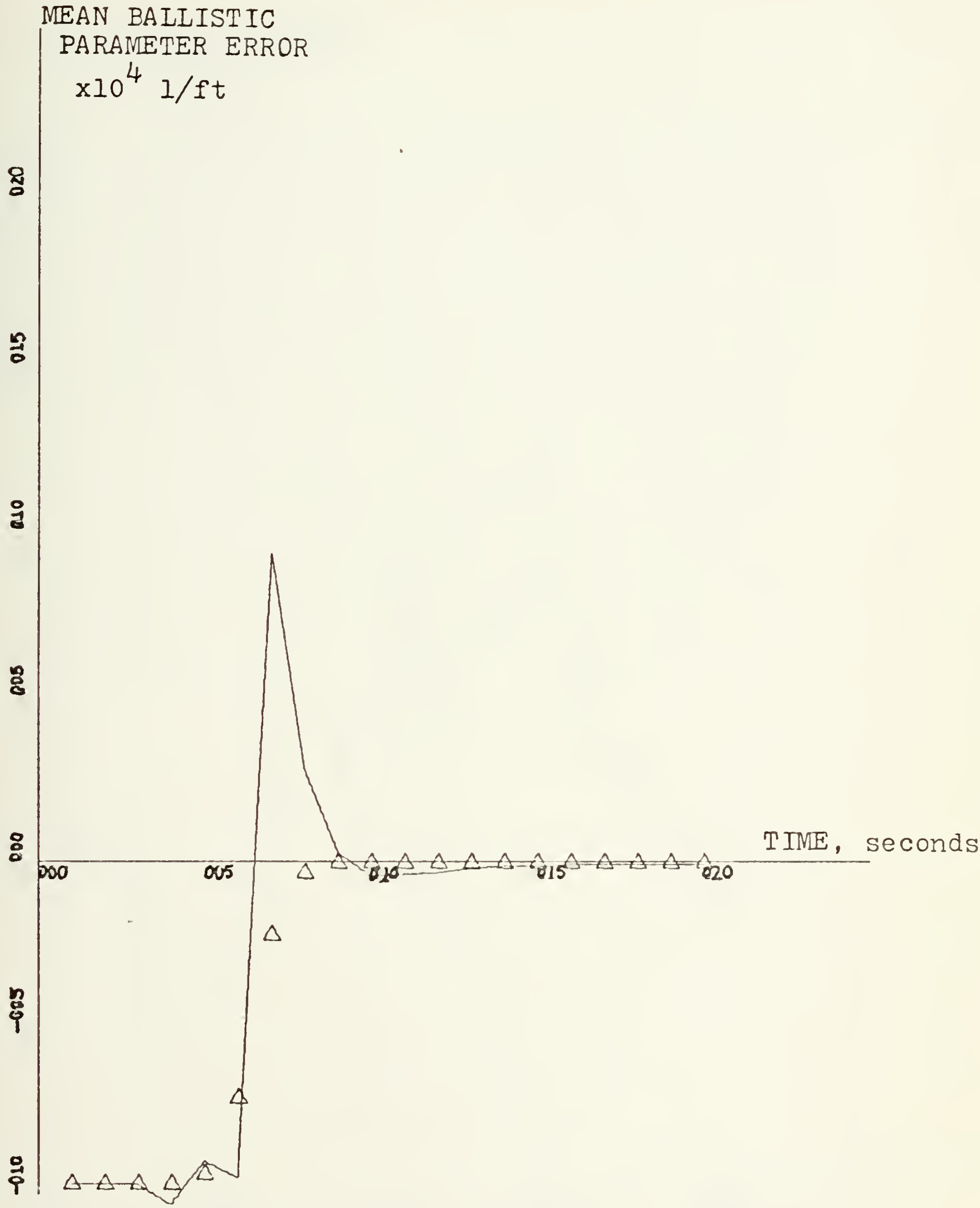

FIG. 5.13. Time history of the mean ballistic parameter estimation error for 50 Monte-Carlo runs.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=5.00E-04 UNITS INCH.

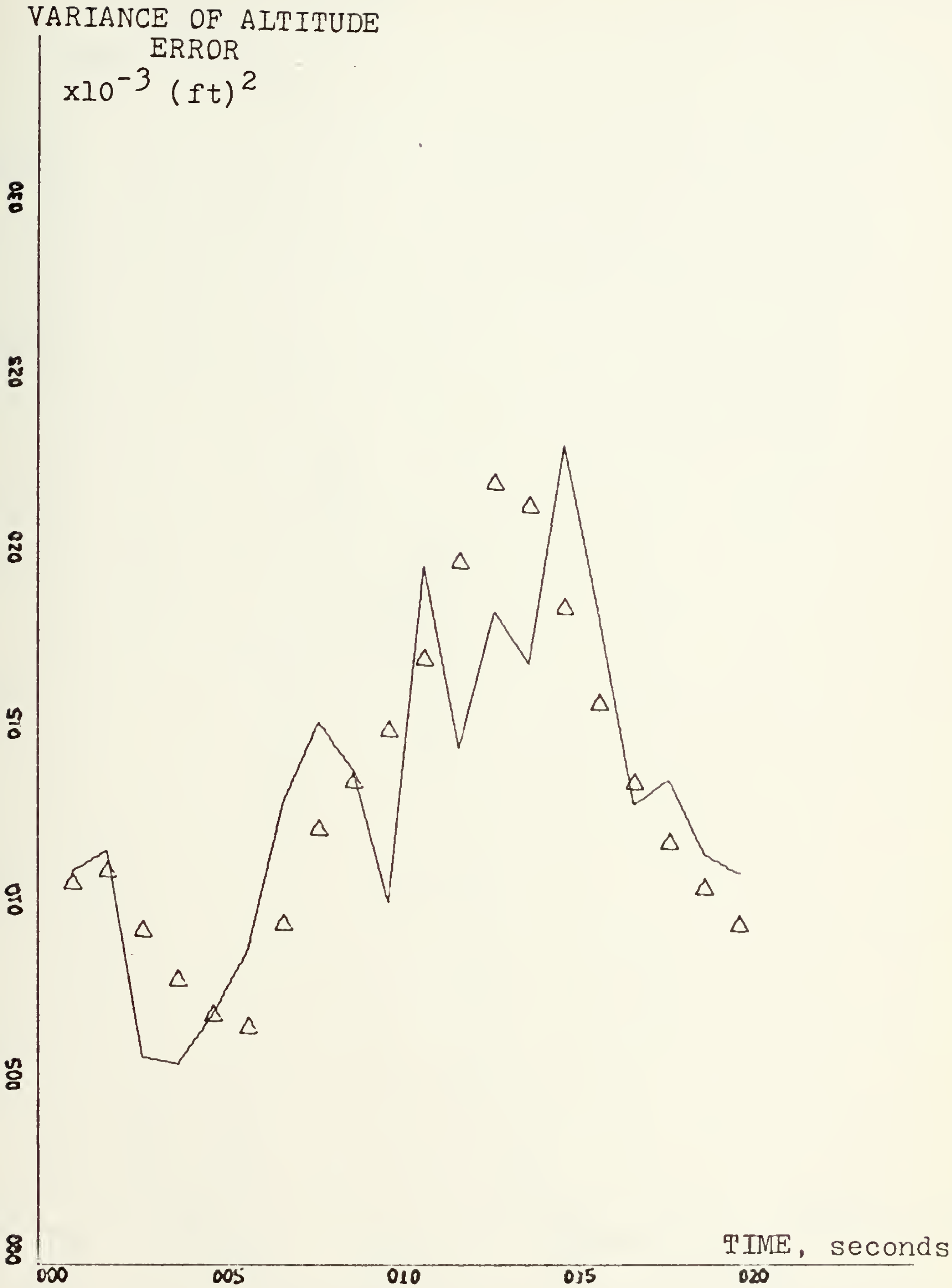

FIG. 5.14. Variance of the altitude estimation error vs. time for 50 Monte-Carlo runs.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=5.00E+03 UNITS INCH.

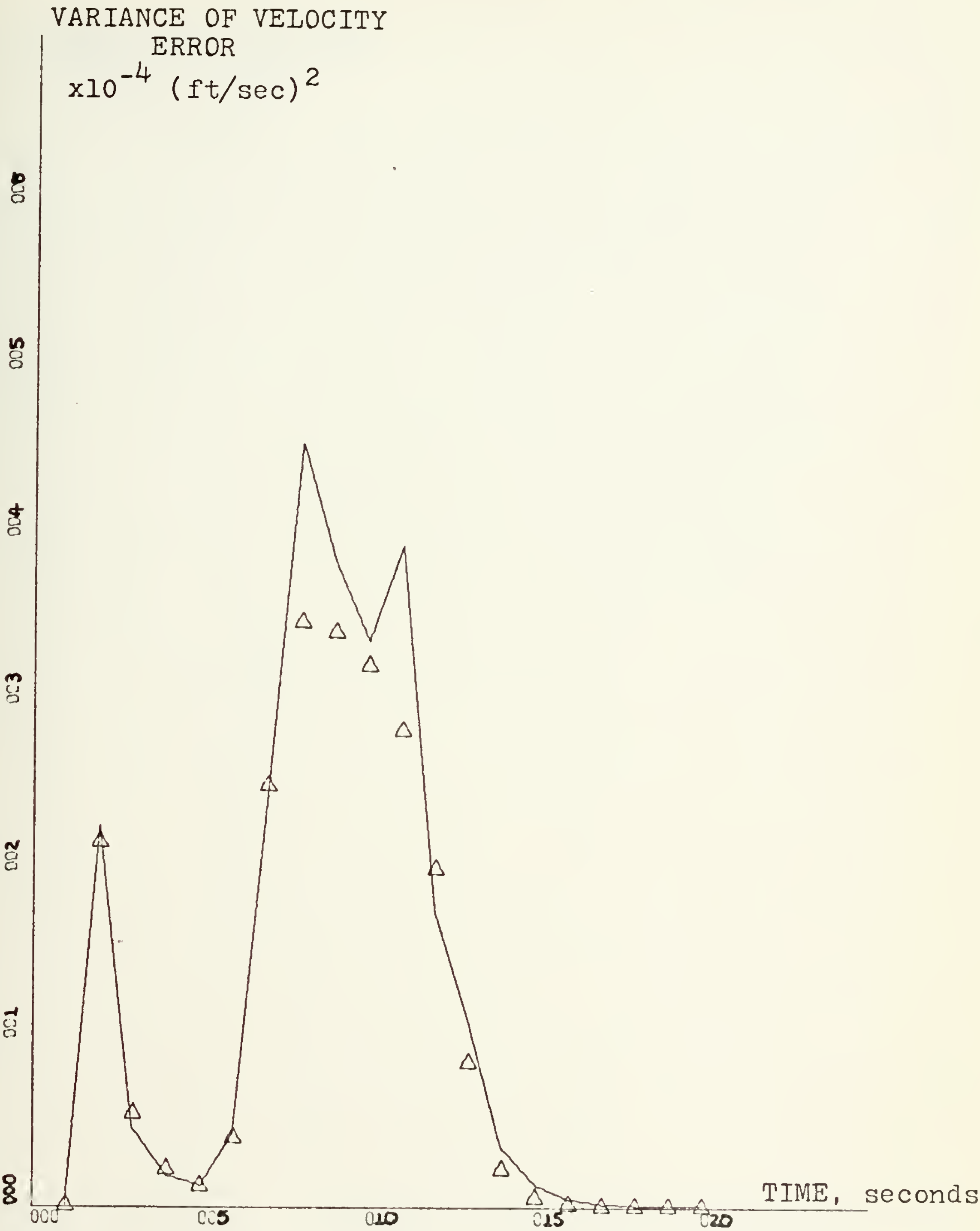

FIG. 5.15. Variance of the velocity estimation error vs. time for 50 Monte-Carlo runs

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=1.00E+04 UNITS INCH.

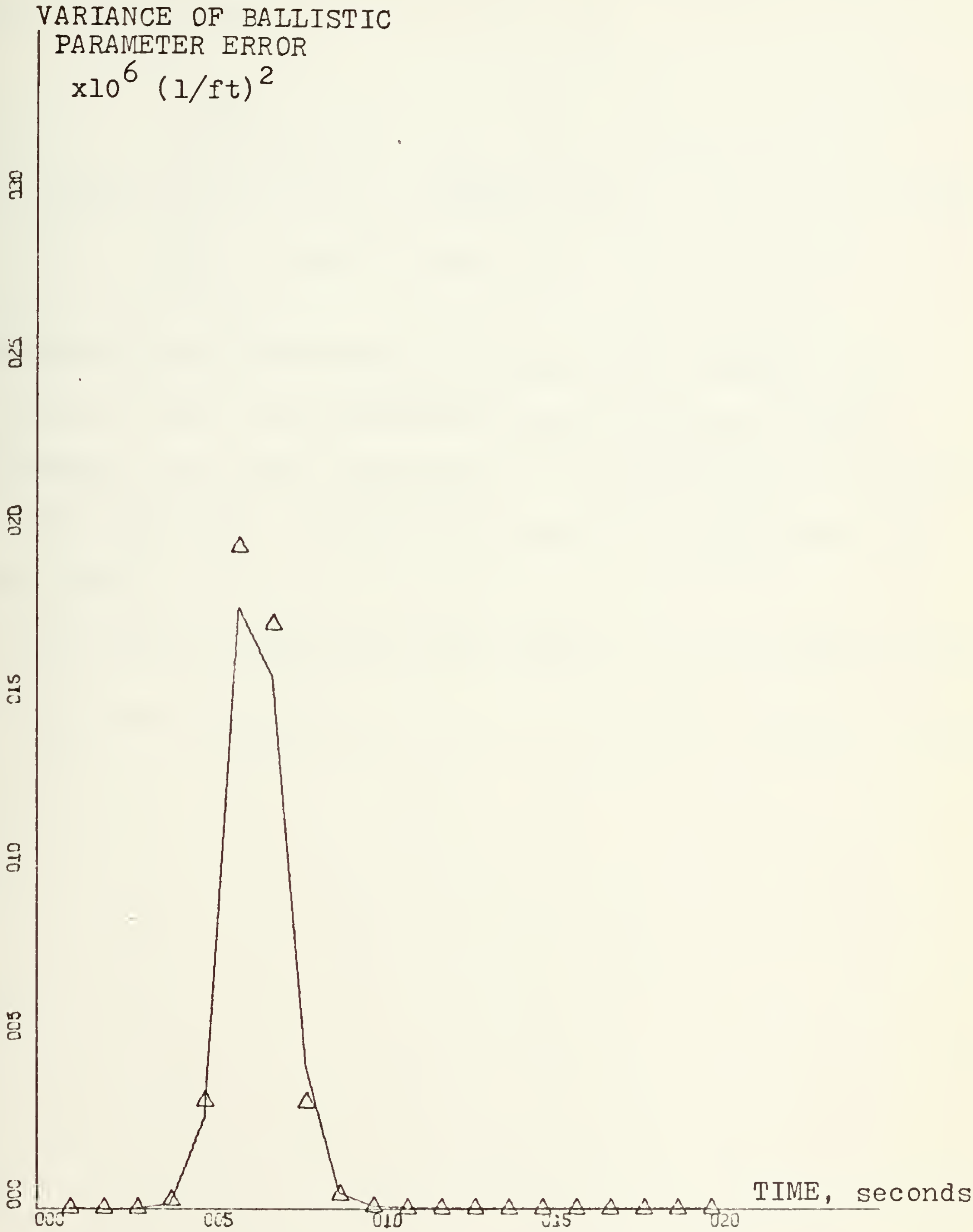

FIG. 5.16. Variance of the ballistic parameter estimation error vs. time for 50 Monte-Carlo runs.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=5.00E-06 UNITS INCH.

Next, the problem was solved for a different target track with initial conditions given by

$$\underline{X}(0) = \begin{bmatrix} 3.5 \times 10^{5} & \text{ft.} \\ 2 \times 10^{4} & \text{ft/sec} \\ 1 \times 10^{-3} & \text{1/ft} \end{bmatrix} = \underline{b} \tag{5.35}$$

The results are illustrated in Figures 517 through 5.22. One can see that the analytical equations predict worse performance than that predicted by the Monte-Carlo simulation.

For comparison, the CPU times used for both method are given below.

Monte-Carlo simulation (1000 runs)...........3 min 21 sec

Analytical equations.........................0 min 9 sec

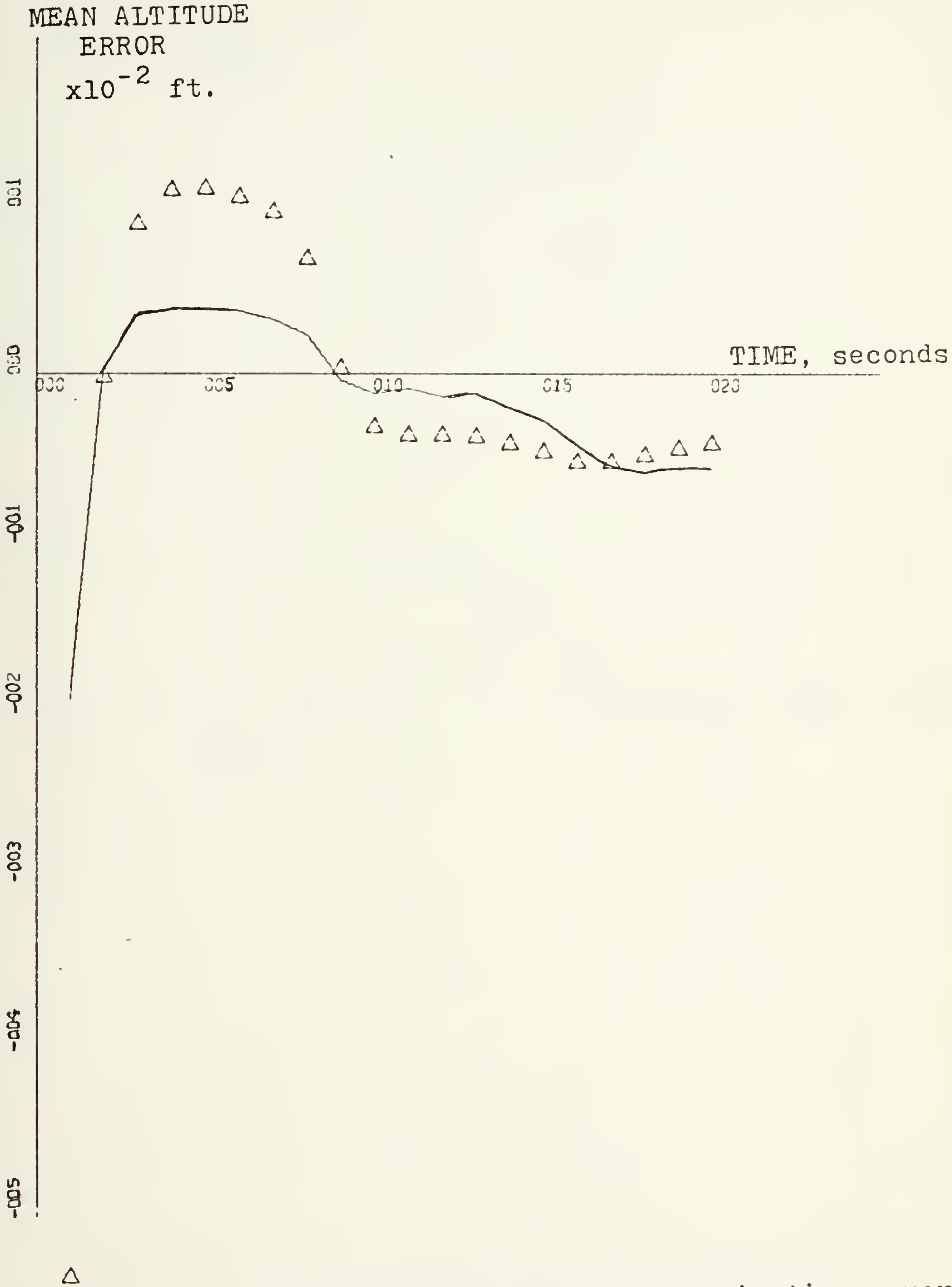

FIG. 5.17. Time history of altitude estimation error for $\underline{X}(0) = \underline{b}$

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=1.00E+02 UNITS INCH.

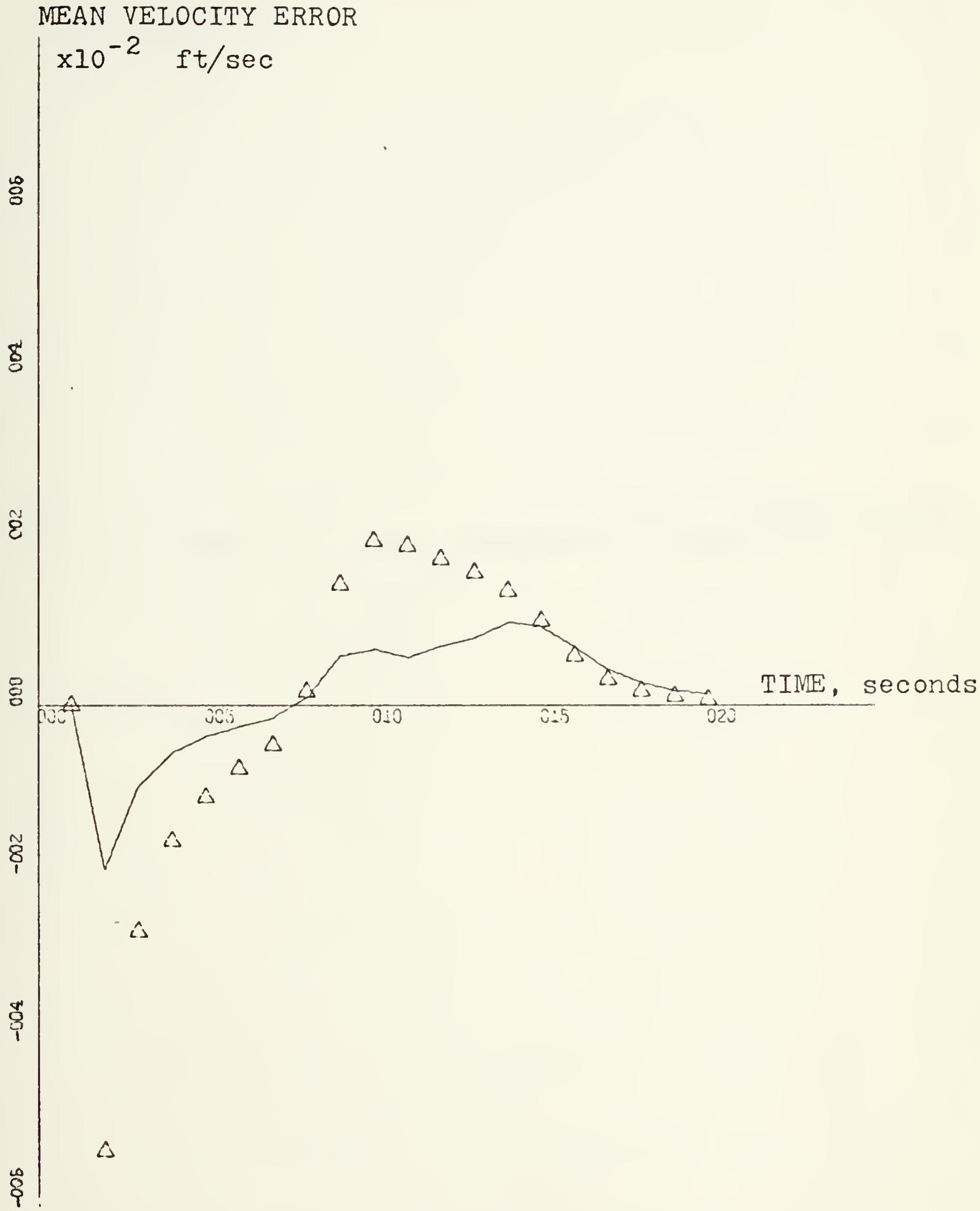

FIG. 5.18. Time history of mean velocity estimation error for $\underline{X}(0)=\underline{b}$.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=2.00E+02 UNITS INCH.

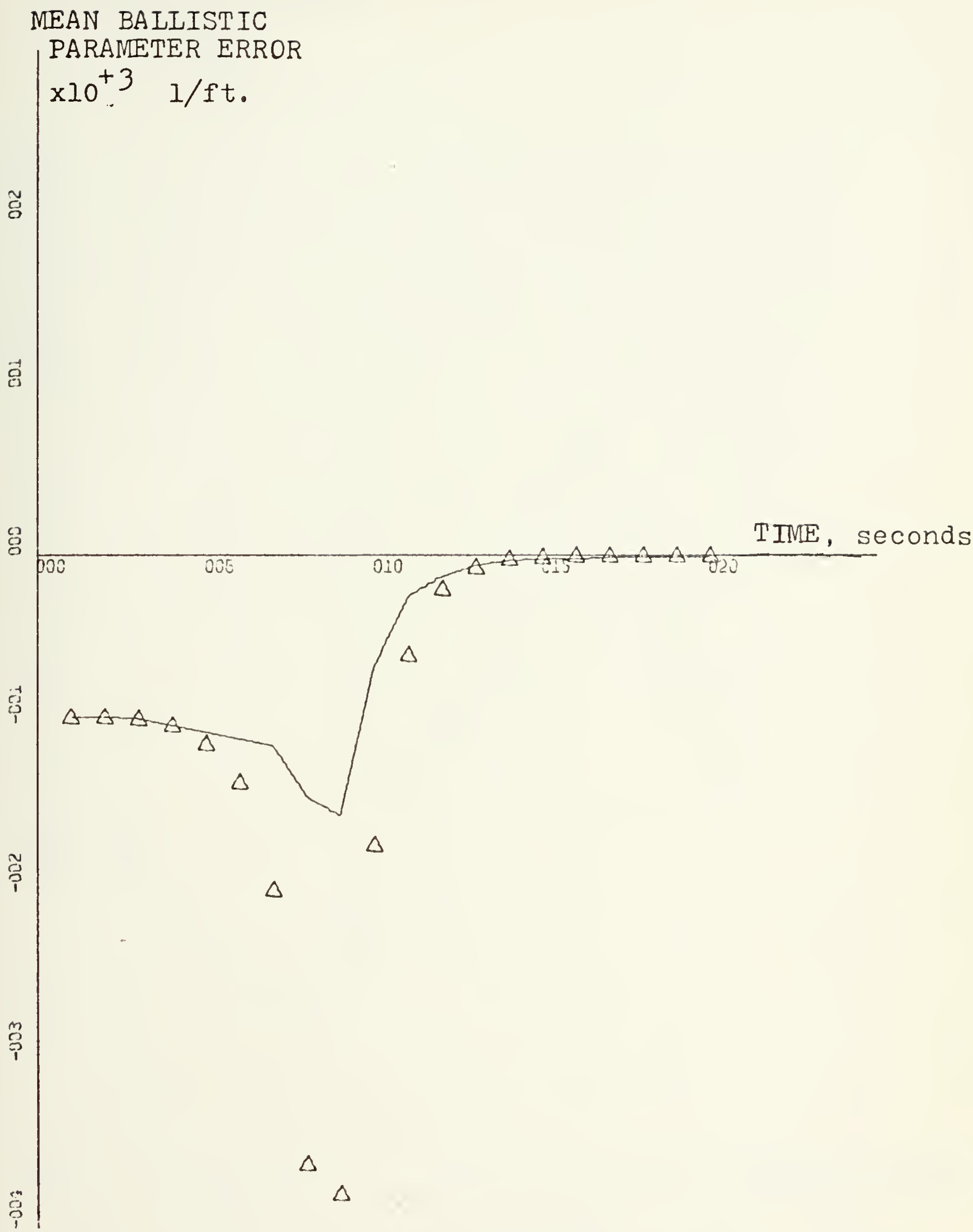

FIG. 5.19. Time history of the ballistic parameter estimation error for $\underline{X}(0) = \underline{b}$

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=1.00E-03 UNITS INCH.

FIG. 5.20. Variance of altitude estimation error vs. time for X(0)=b.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=5.00E+03 UNITS INCH.

FIG. 5.21. Variance of velocity estimation error vs. time for X(0)=b.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=5.00E+03 UNITS INCH.

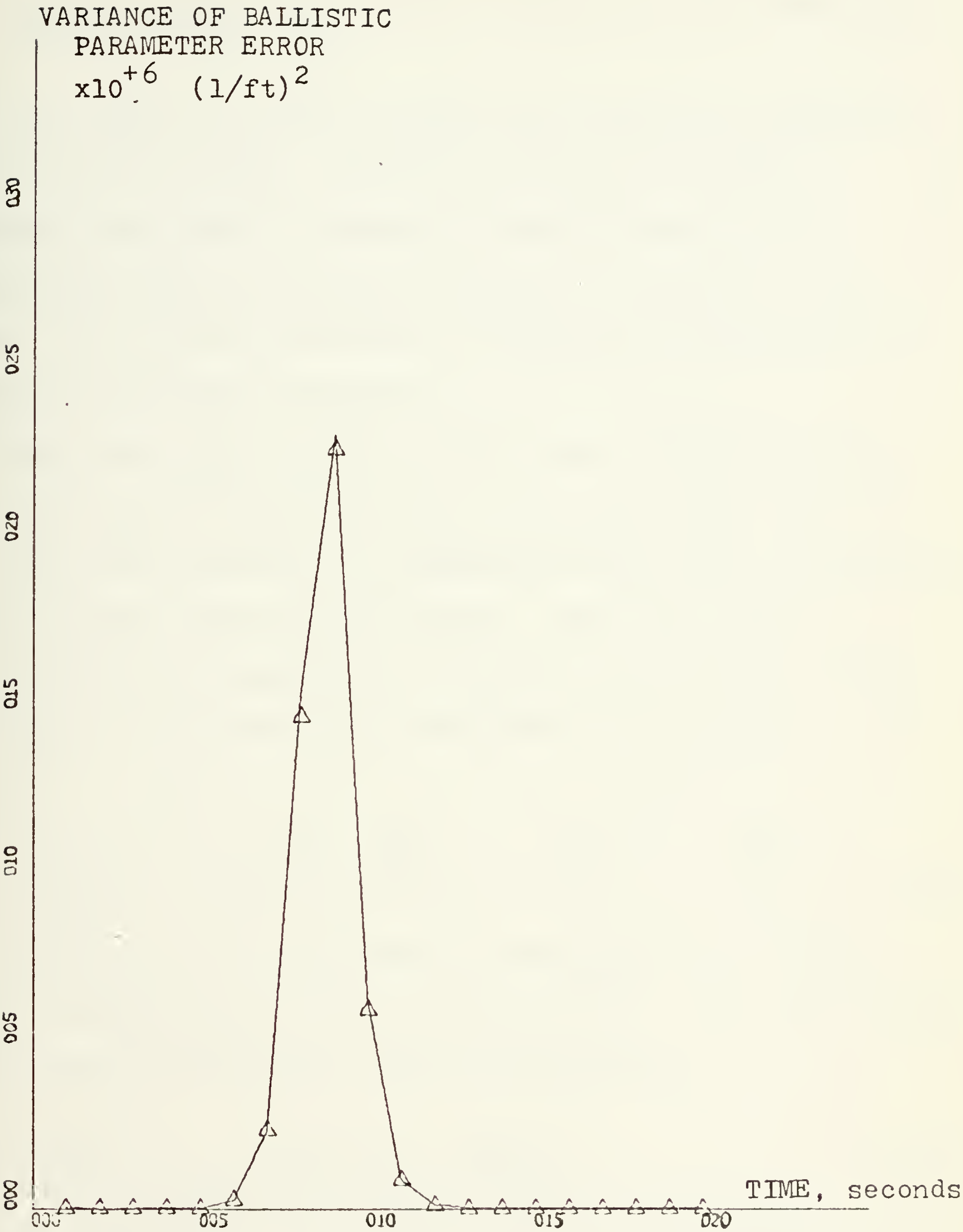

FIG. 5.22. Variance of ballistic parameter estimation error vs. time for X(0)=b.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=5.00E-06 UNITS INCH.

B. APPLICATION OF THE ANALYTICAL APPROACH TO THE MULTIPLE TRACK NONLINEAR CASE

The results derived in Chapter III can also be applied to nonlinear problems. Equation (3.11) can be used to calculate the mean of estimation error in terms of the conditional means of estimation error calculated for each track. The conditional means are computed approximately by using the analytical equations.

Since the matrices $\underset{\sim}{\emptyset}(k)$ and $\underset{\sim}{H}(k)$ are calculated using the true target trajectory in the nonlinear case, one can see that the covariance of estimation error will be dependent on the tracks (see Equation (2.31). Thus, one needs a relationship to compute the covariance of estimation error in terms of the conditional covariances of estimation error, i.e.

$$\underset{\sim}{P}(k/k) = f\left(\overset{(1)}{\underset{\sim}{P}}(k/k), \overset{(2)}{\underset{\sim}{P}}(k/k), \ldots, \overset{(i)}{\underset{\sim}{P}}(k/k), \ldots, \overset{(n)}{\underset{\sim}{P}}(k/k) \right) \tag{5.36}$$

where $\overset{(i)}{\underset{\sim}{P}}(k/k)$ is the conditional covariance of estimation error corresponding to the i'th track $\overset{(i)}{\underline{X}}(k)$. The conditional covariance of estimation error is defined as

$$\overset{(i)}{\underset{\sim}{P}}(k/k) = \text{Covar}\left[\tilde{\underline{X}}(k)/\underline{X}(k)=\overset{(i)}{\underline{X}}(k) \right] \tag{5.37}$$

The relationship of the covariance to the conditional covariances is derived in the following paragraphs.

C. CONDITIONAL COVARIANCE

Given a discrete random vector $\underline{B}$, the expected value of the dependent continuous random vector $\underline{A}$ can be expressed in terms of the conditional expectations as

$$E[\underline{A}] = \sum_j E\left[\underline{A}/\underline{B}=\underline{b}_j\right] \cdot P[\underline{B}=\underline{b}_j] \qquad (5.38)$$

For simplicity Equation (5.38) can be written as

$$E\left[\underline{A}\right] = E\left\{E[\underline{A}/\underline{B}]\right\}_B \qquad (5.39)$$

where the subscript B denotes the expectation with respect to the random vector $\underline{B}$.

The conditional covariance is defined as [3]

$$\underset{\sim}{P}_{A/B} = E\left\{\left[\left(\underline{A} - E[\underline{A}/\underline{B}]\right) \cdot \left(\underline{A} - E[\underline{A}/\underline{B}]\right)^T\right] / \underline{B}\right\} \qquad (5.40)$$

But

$$\underset{\sim}{P}_A = E\left\{[\underline{A} - E\{\underline{A}\}] \cdot [\underline{A} - E\{\underline{A}\}]^T\right\} \qquad (5.41)$$

which is the (unconditional) covariance matrix. Replacing $\underline{A}$ with

$[\underline{A} - E\{\underline{A}\}] \cdot [\underline{A} - E\{\underline{A}\}]^T$ in Equation (5.39) gives **

$$\underset{\sim}{P}_A = E\left\{E\left\{[\underline{A} - E\{\underline{A}\}] \cdot [\underline{A} - E\{\underline{A}\}]^T / \underline{B}\right\}\right\}_B \qquad (5.42)$$

**Equation (5.39) can be written in the general case as $E\{g(a)\} = E\{E[g(a)/A]\}$ where g(a) may be any function.

By adding and subtracting $E\left[\underline{A}/\underline{B}\right]$ inside the parantheses, Equation (5.42) can be written as

$$P_A = E\left\{ E\left\{\left[\left(\underline{A} - E\left[\underline{A}/\underline{B}\right]\right) + \left(E\left[\underline{A}/\underline{B}\right] - E\left[\underline{A}\right]\right)\right] \cdot \right.\right.$$

$$\left.\left.\left[\left(\underline{A} - E\left[\underline{A}/\underline{B}\right]\right) + \left(E\left[\underline{A}/\underline{B}\right] - E\left[\underline{A}\right]\right)\right]^T / \underline{B}\right\}\right\}_B$$

or

$$\begin{aligned} P_A = E\Big\{ E\Big\{\Big[& (\underline{A} - E[\underline{A}/\underline{B}]) \cdot (\underline{A} - E[\underline{A}/\underline{B}])^T \\ & + (\underline{A} - E[\underline{A}/\underline{B}]) \cdot (E[\underline{A}/\underline{B}] - E[\underline{A}])^T + (E[\underline{A}/\underline{B}] - E[\underline{A}]) \\ & \cdot (\underline{A} - E[\underline{A}/\underline{B}])^T + (E[\underline{A}/\underline{B}] - E[\underline{A}]) \\ & \cdot (E[\underline{A}/\underline{B}] - E[\underline{A}])^T\Big] / \underline{B}\Big\}\Big\}_B \end{aligned} \qquad (5.43)$$

Using the properties of the expectation operator in the inner expectation, the cross terms can be calculated as

$$\begin{aligned} E\left\{(\underline{A} - E[\underline{A}/\underline{B}]) \cdot (E[\underline{A}/\underline{B}] - E[\underline{A}])^T/\underline{B}\right\} = {} & E\left\{(\underline{A} - E[\underline{A}/\underline{B}])/\underline{B}\right\} \\ & \cdot (E[\underline{A}/\underline{B}] - E[\underline{A}])^T \end{aligned} \qquad (5.44)$$

(because $E[\underline{A}]$ and $E[\underline{A}/\underline{B}]$ are not functions of $\underline{A}$, they are deterministic with respect to $\underline{A}$ for a given value of $\underline{B}$. This means $E_A\left\{E[\underline{A}/\underline{B}]\right\} = E[\underline{A}]$ and $E_A\left\{E[\underline{A}/\underline{B}]/\underline{B}\right\} = E_A\left\{\underline{A}/\underline{B}\right\}$).

Using the properties of the expectation operator yields

$$E\left\{ \left(\underline{A} - E\left[\underline{A}/\underline{B}\right]\right)/\underline{B} \right\} = E\left[\underline{A}/\underline{B}\right] - E\left\{ E\left[\underline{A}/\underline{B}\right]/\underline{B}\right\}$$

$$= \left(E\left[\underline{A}/\underline{B}\right] - E\left[\underline{A}/\underline{B}\right]\right)$$

$$= \underline{0} \qquad (5.45)$$

Thus, the cross terms drop out in Equation (5.43). Since $\left(E\left[\underline{A}/\underline{B}\right] - E\left[\underline{A}\right]\right)$ is not a function of $\underline{A}$ but it is a function of $\underline{B}$, then Equation (5.43) reduces to

$$\underset{\sim}{P}_A = E\left\{ E\left\{ \left(\underline{A} - E\left[\underline{A}/\underline{B}\right]\right)\cdot\left(\underline{A} - E\left[\underline{A}/\underline{B}\right]\right)^T/\underline{B}\right\} + \left(E\left[\underline{A}/\underline{B}\right] - E\left[\underline{A}\right]\right)\cdot\left(E\left[\underline{A}/\underline{B}\right] - E\left[\underline{A}\right]\right)^T \right\}_B \qquad (5.46)$$

or

$$\underset{\sim}{P}_A = E\left\{ E\left\{ \left(\underline{A} - E\left[\underline{A}/\underline{B}\right]\right)\cdot\left(\underline{A} - E\left[\underline{A}/\underline{B}\right]\right)^T/\underline{B}\right\}\right\}_B + E\left\{ \left(E\left[\underline{A}/\underline{B}\right] - E\left[\underline{A}\right]\right)\cdot\left(E\left[\underline{A}/\underline{B}\right] - E\left[\underline{A}\right]\right)^T \right\}_B \qquad (5.47)$$

The inner expectation in the first term of the right hand side defines the conditional covariance (see Equation (5.40), hence

$$\underset{\sim}{P}_A = E\left\{ \underset{\sim}{P}_{A/B} \right\}_B + E\left\{ \left(E\left[\underline{A}/\underline{B}\right] - E\left[\underline{A}\right]\right)\cdot\left(E\left[\underline{A}/\underline{B}\right] - E\left[\underline{A}\right]\right)^T\right\}_B \qquad (5.48)$$

Equation (5.48) gives the required relationship between the covariance and conditional covariances, i.e.

$$\underset{\sim}{P}(k/k) = E\left\{ \text{Covar}\left\{ \underline{\tilde{X}}(k)/\underline{X}(k)=\overset{(i)}{\underline{X}}(k)\right\} \right\}_{\underline{X}(k)}$$

$$+ E\left\{ \left(\overset{(i)}{\underline{\mu}}(k) - \underline{\mu}(k) \right) \cdot \left(\overset{(i)}{\underline{\mu}}(k) - \underline{\mu}(k) \right)^T \right\}_{\underline{X}(k)}$$

$$= E\left\{ \overset{(i)}{P}(k/k) \right\}_{\underline{X}(k)} + E\left\{ \left(\overset{(i)}{\underline{\mu}}(k) - \underline{\mu}(k) \right) \right.$$

$$\left. \cdot \left(\overset{(i)}{\underline{\mu}}(k) - \underline{\mu}(k) \right)^T \right\}_{\underline{X}(k)}$$

which can also be expressed as

$$\underset{\sim}{P}(k/k) = \sum_{i=1}^{n} \overset{(i)}{P}(k/k) \cdot p_i + \sum_{i=1}^{n} \left[\overset{(i)}{\underline{\tilde{\mu}}}(k) - \underline{\tilde{\mu}}(k) \right]$$

$$\cdot \left[\overset{(i)}{\underline{\tilde{\mu}}}(k) - \underline{\tilde{\mu}}(k) \right]^T \cdot p_i \qquad (5.49)$$

where

$\overset{(i)}{\underline{\tilde{\mu}}}(k)$ and $\overset{(i)}{\underset{\sim}{P}}(k/k)$ are the conditional mean and covariance of estimation error for the i'th track $\overset{(i)}{\underline{X}}(k)$,

$\underline{\tilde{\mu}}(k)$ and $\underset{\sim}{P}(k/k)$ are the overall mean and covariance of estimation error and

p_i is the probability of occurrence of the i'th track $\overset{(i)}{\underline{X}}(k)$.

First, using Equation (3.11) one can compute the mean of estimation error $\underline{\mu}(k)$. Then using Equation (5.49) the covariance can be calculated.

The re-entry problem was simulated for two tracks where

$$p_1 = P\left\{\underline{X}(k)=\overset{(1)}{\underline{X}}(k)\right\}$$

$$= 0.5 \quad (5.50)$$

$$p_2 = P\left\{\underline{X}(k)=\overset{(2)}{\underline{X}}(k)\right\}$$

$$= 0.5 \quad (5.51)$$

with

$$\overset{(1)}{\underline{X}}(0)=\begin{bmatrix} 3 \times 10^{5} & \text{ft.} \\ 2 \times 10^{4} & \text{ft/sec} \\ 1 \times 10^{-3} & \text{1/ft} \end{bmatrix} \quad (5.52)$$

$$\overset{(2)}{\underline{X}}(0)=\begin{bmatrix} 3.5 \times 10^{5} & \text{ft.} \\ 2 \times 10^{4} & \text{ft/sec} \\ 1 \times 10^{-3} & \text{1/ft} \end{bmatrix} \quad (5.53)$$

Figures 5.23 through 5.28 illustrate the results. The continuous curves represent the Monte-Carlo simulation and the triangles represent the results from the analytical equations. From the figures one can see that the 1000-member ensemble Monte-Carlo simulation predicts better performance. Actually the differences between the two results are small compared to the values of the state variables. For example, the maximum difference between the two results is 150 feet for

the altitude error at 300,000 feet altitude (which is the initial difference; at later times the two results are closer). The CPU times used for these simulations are given below.

	CPU Time
Monte-Carlo simulation (1000 runs).......	3 min 50 sec
Monte-Carlo simulation (50 runs).........	0 min 17 sec
Analytical equations.....................	0 min 7 sec

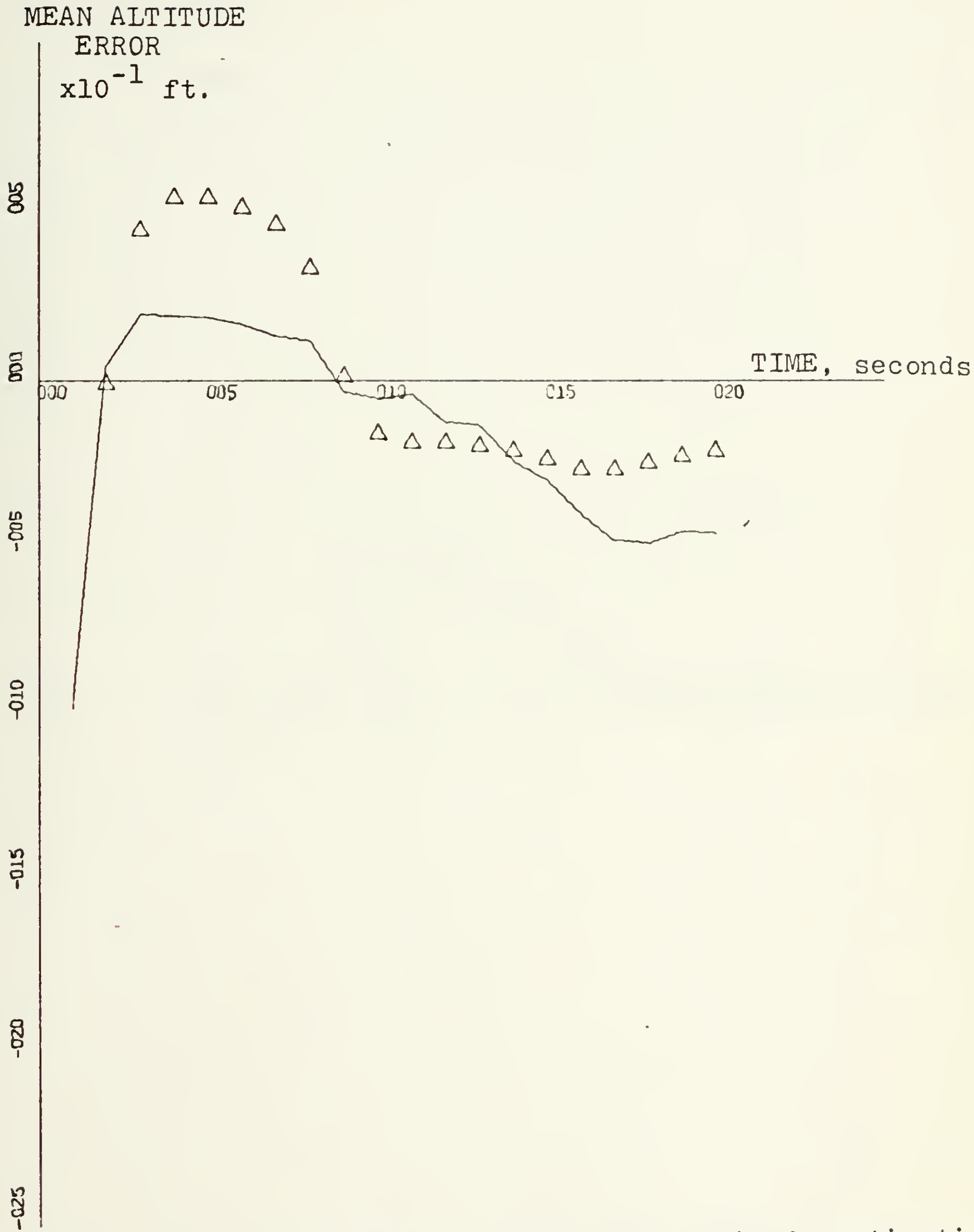

FIG. 5.23. Time history of the mean altitude estimation error for two tracks.
△

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=5.00E+01 UNITS INCH.

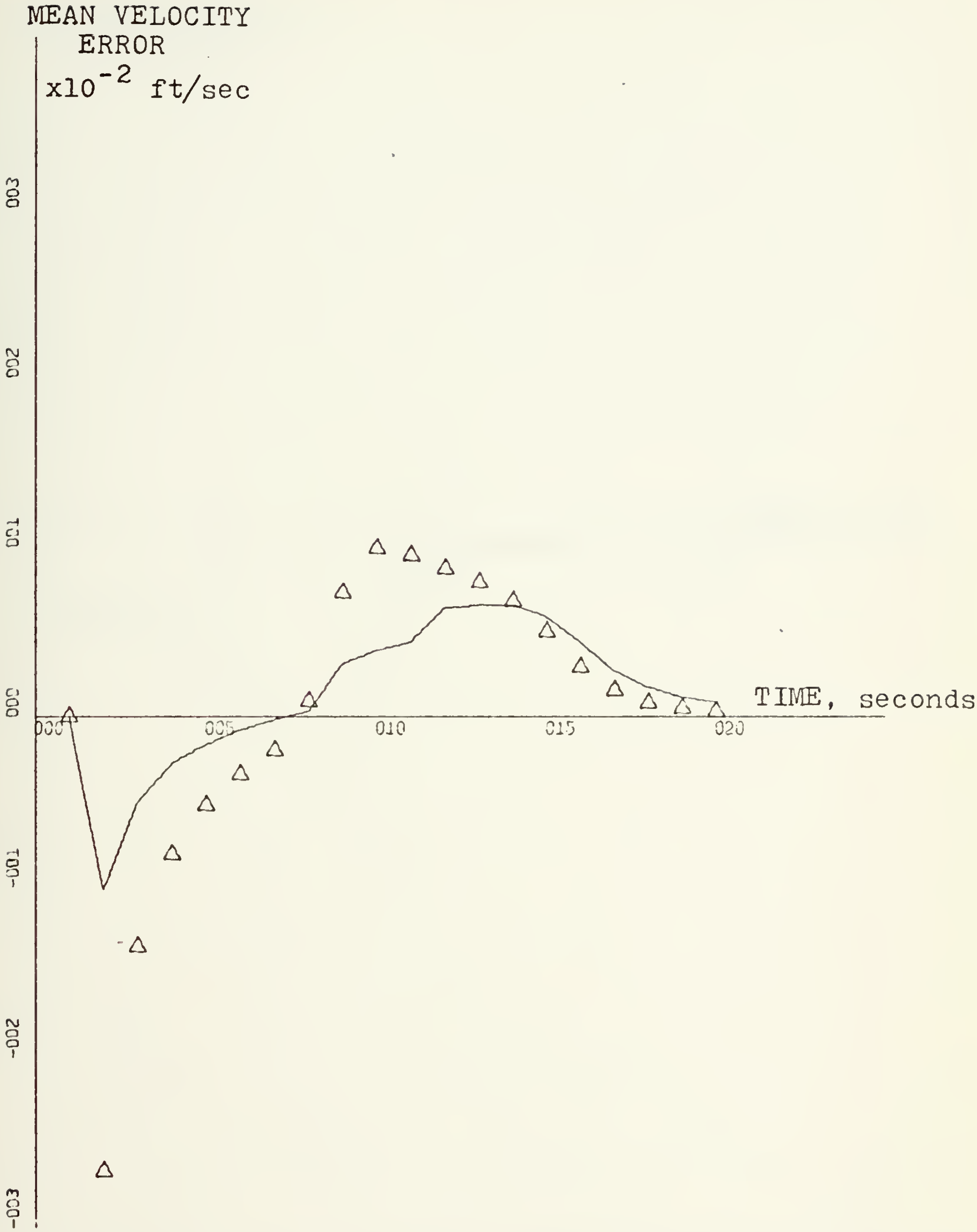

FIG. 5.24. Time history of the mean velocity estimation error for two tracks.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=1.00E+02 UNITS INCH.

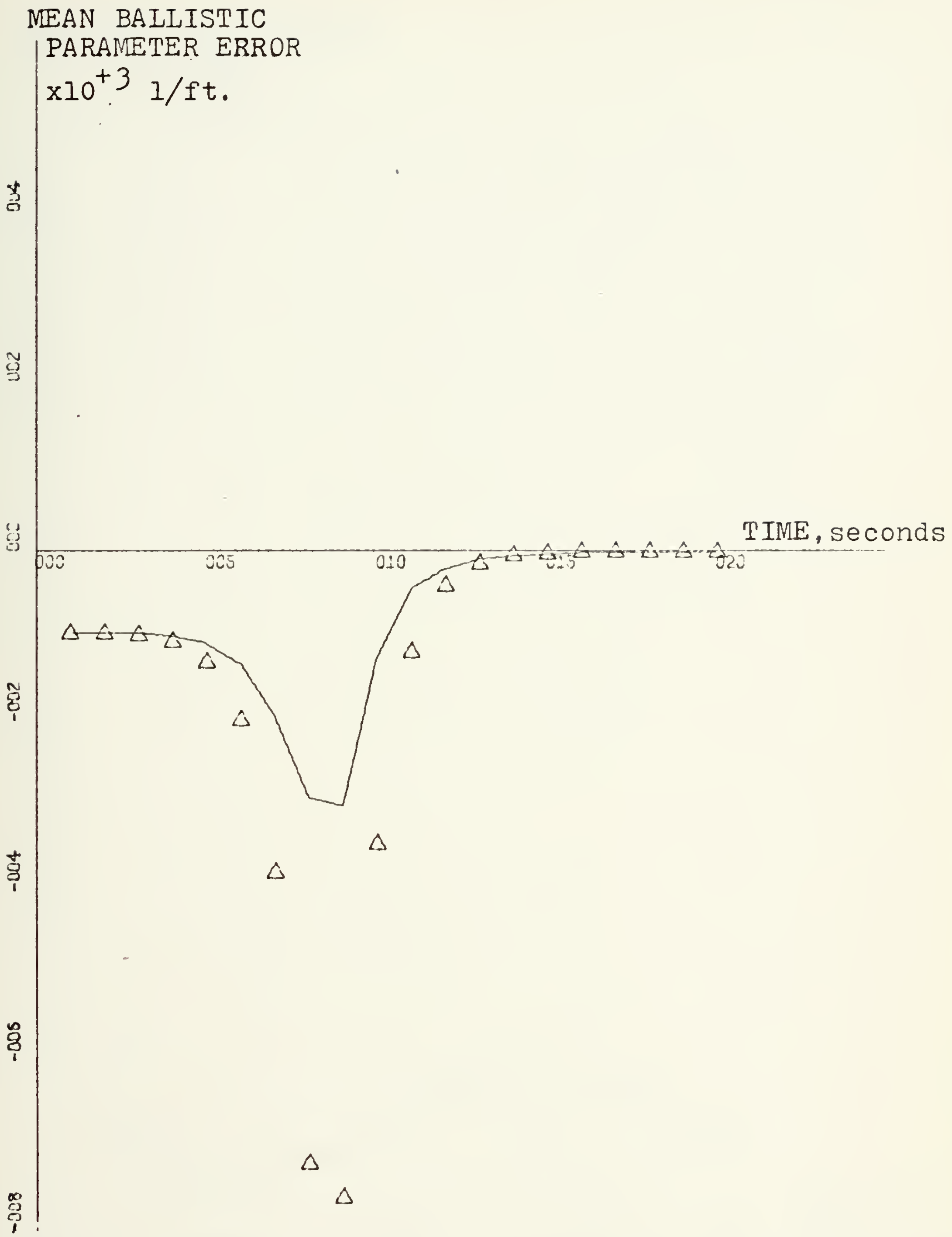

FIG. 5.25. Time history of the mean ballistic parameter estimation error for two tracks.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=2.00E-03 UNITS INCH.

FIG. 5.26. Time history of the variance of the altitude estimation error for two tracks.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=2.00E+04 UNITS INCH.

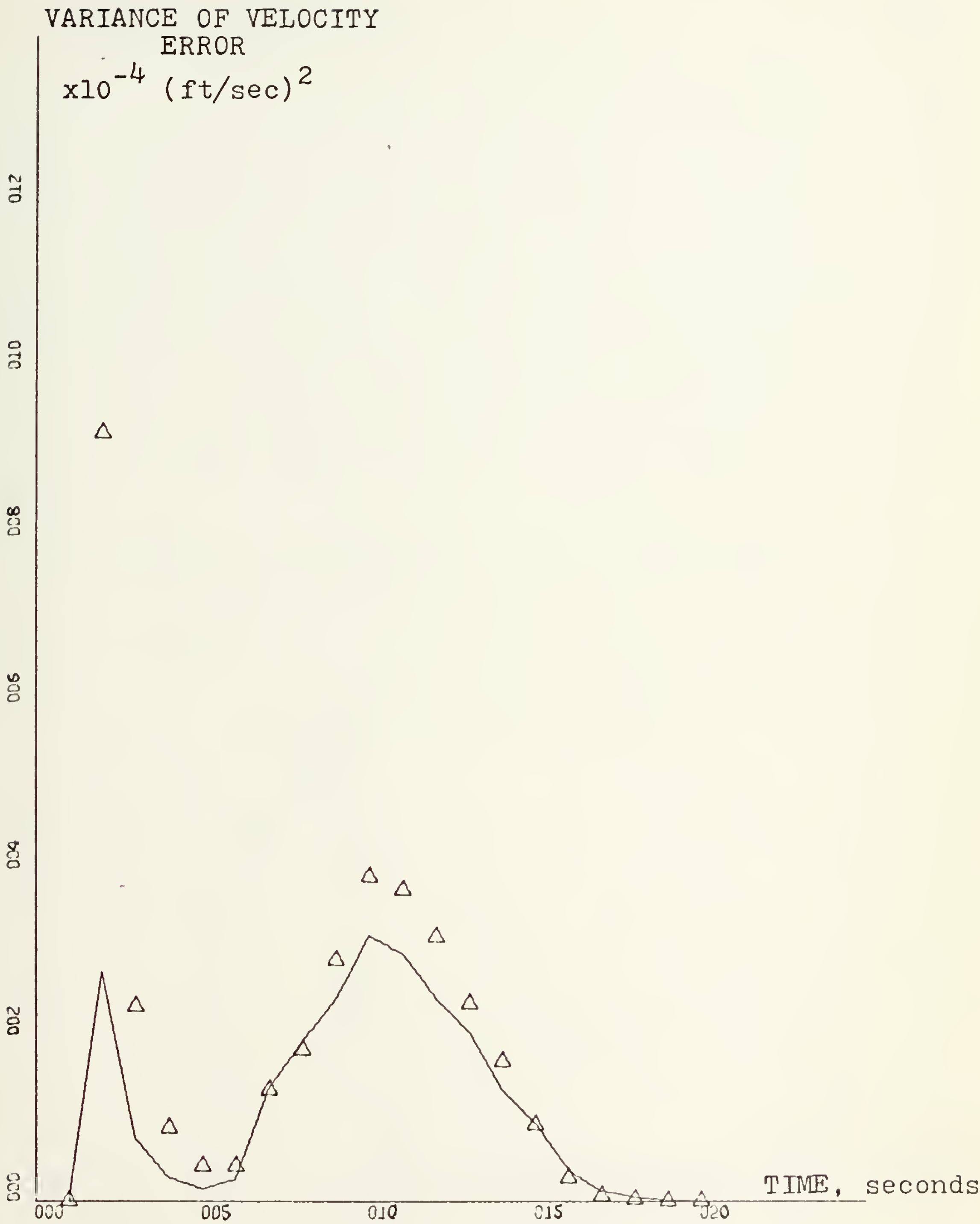

FIG. 5.27. Time history of the variance of the velocity estimation error for two tracks.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=2.00E+04 UNITS INCH.

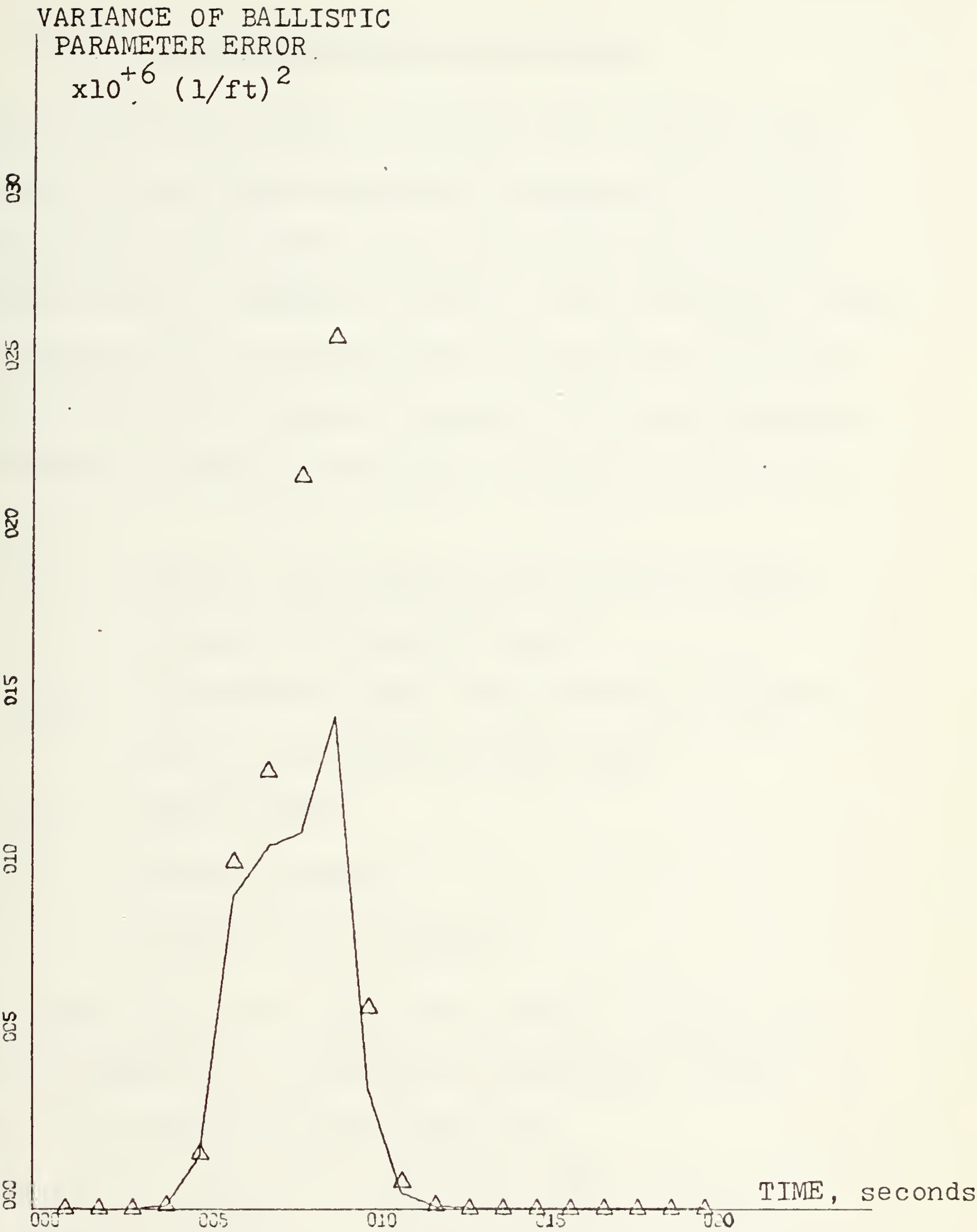

FIG. 5.28. Time history of the variance of the ballistic parameter estimation error for two tracks.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=5.00E-06 UNITS INCH.

VI. A SUBMARINE TRACKING PROBLEM

This chapter discusses an example with linear state equations but nonlinear measurement equations. It is a problem in which estimates are made of the states of a submerged target: heading, velocity, rest frequency, range to the target at the closest point of approach (cpa) and distance to cpa. One passive sensor (sono bouy) provides measurements of target heading and frequency [4]. The states are

R_{cpa} : Range to the target at the closest point of approach from the sensor.

X_{cpa} : X distance to cpa from the sensor (negative before cpa, positive after cpa).

V_s : Target speed.

θ_s : Target heading.

F_o : Target rest frequency.

It is assumed that the target has constant velocity and heading. Figure 6.1 illustrates the geometry of the problem. The state equations are [4]

$$R_{cpa}(k+1) = R_{cpa}(k) + g_1 (\gamma_{\dot{V}_s}, \gamma_{\dot{\theta}_s}, k)$$

$$X_{cpa}(k+1) = X_{cpa}(k) + V_s \cdot T + g_2 (\gamma_{\dot{V}_s}, \gamma_{\dot{\theta}_s}, k)$$

$$V_s(k+1) = V_s(k) + g_3(\gamma_{\dot{V}_s}) \tag{6.1}$$

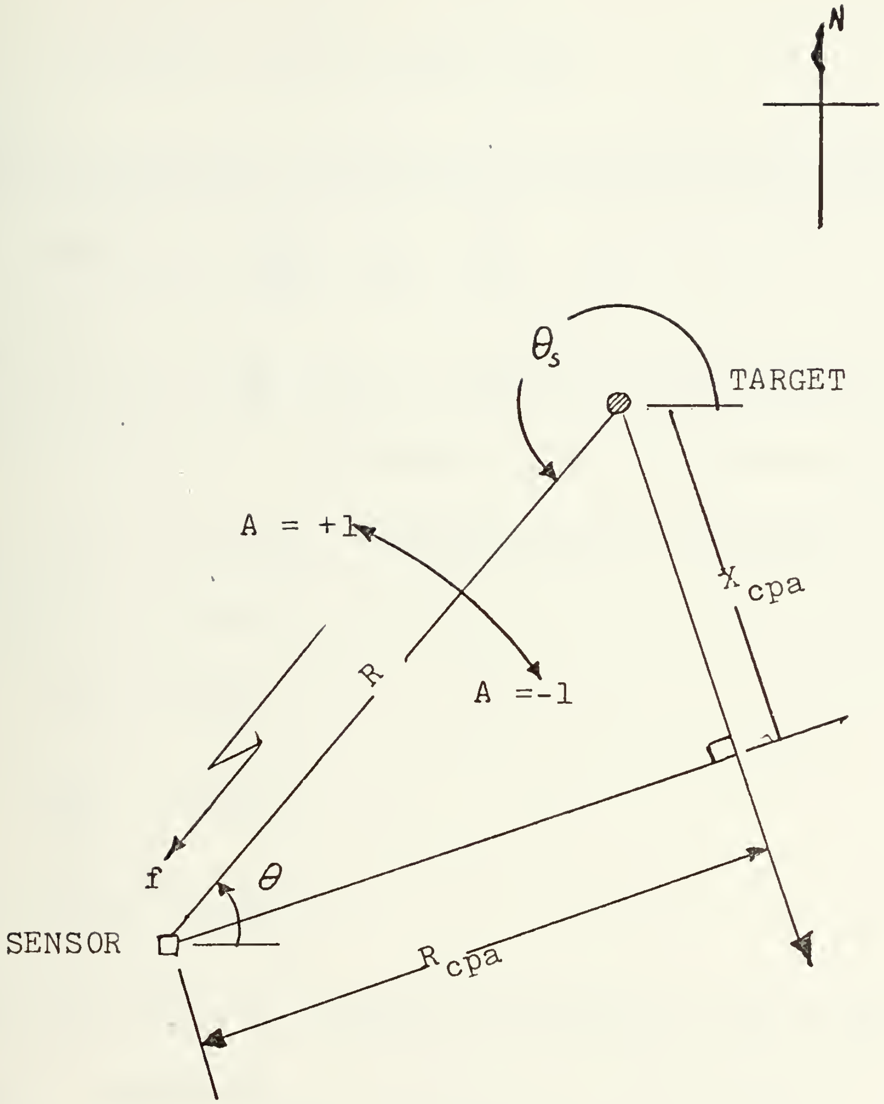

FIG. 6.1. R_{cpa} - X_{cpa} filter geometry.

$$_s(k+1) = \theta_s(k) + g_4(\gamma_{\dot{\theta}_s})$$

$$F_o(k+1) = F_o(k) + g_5(\gamma_{\dot{F}_o})$$

with g_1 through g_5 the random forcing terms, hence,

$$\underline{W}(k) = \underline{g}(\gamma_{\dot{\theta}_s}, \gamma_{\dot{V}_s}, \gamma_{\dot{F}_o}, k) \tag{6.2}$$

The quantities $\gamma_{\dot{\theta}_s}$, $\gamma_{\dot{V}_s}$, $\gamma_{\dot{F}_o}$ are random changes in heading, velocity and rest frequency. They are assumed to be independent, zero mean, piecewise-constant rates of change. They have variances defined by

$$\Sigma_{\dot{V}_s}^2 = E\{\gamma_{V_s}^2\}$$

$$\Sigma_{\dot{\theta}_s}^2 = E\{\gamma_{\theta_s}^2\}$$

$$\Sigma_{\dot{F}_o}^2 = E\{\gamma_{F_o}^2\}$$

The values for the standard deviations were taken from typical maneuvering parameters for the target

$$\Sigma_{\dot{\theta}_s} = 100^\circ / 1000 \text{ sec} = 1.74533 \times 10^{-3} \text{ rad/sec}$$

$$\Sigma_{\dot{V}_s} = 10 \text{ kts} / 1000 \text{ sec} = 5.5555 \times 10^{-3} \text{ yds/sec}^2 \tag{6.3}$$

$$\Sigma_{\dot{F}_o} = 0.5 \text{ Hz}/ 1000 \text{ sec} = 0.5 \times 10^{-3} \text{ Hz/sec}$$

With the expressions for the random forcing terms included, the state equation become [4]

$$R_{cpa}(k+1) = R_{cpa}(k) + A \cdot X_{cpa}(k) \cdot \gamma_{\dot{\theta}_s} \cdot T$$

$$X_{cpa}(k+1) = X_{cpa}(k) + V_s(k) \cdot T + \frac{1}{2} \cdot \gamma_{\dot{V}_s}$$

$$\cdot T^2 - A \cdot R_{cpa}(k) \cdot \gamma_{\theta_s} \cdot T$$

$$V_s(k+1) = V_s(k) + \gamma_{\dot{V}_s} \cdot T \qquad (6.4)$$

$${}_s(k+1) = {}_s(k) + \gamma_{\dot{\theta}_s} \cdot T$$

$$F_o(k+1) = F_o(k) + \gamma_{\dot{F}_o} \cdot T$$

where A is +1 for counter-clockwise rotation about the sensor and -1 for clockwise rotation about the sensor. A is needed to give the correct sign for a given geometry.

The angle measurement equation is

$$\theta(k) = \theta_s(k) - \mathrm{Tan}^{-1}\left[\frac{A \cdot R_{cpa}(k)}{X_{cpa}(k)}\right] + V_\theta(k) \quad \text{for } X_{cpa} < 0$$

$$(6.5)$$

$$= \theta_s(k) - \mathrm{Tan}^{-1}\left[\frac{A \cdot R_{cpa}(k)}{X_{cpa}(k)}\right] - 180^\circ + V_\theta(k) \quad \text{for } X_{cpa} > 0$$

The frequency observation equation is

$$f(k) = \frac{F_o(k) \cdot V_p}{V_p + \dfrac{V_s(k) \cdot X_{cpa}(k)}{\sqrt{R_{cpa}(k)^2 + X_{cpa}(k)^2}}} + V_f(k) \qquad (6.6)$$

where $V_\theta(k)$, $V_f(k)$ are measurement noises and V_p is the velocity of sound in the medium (V_p is assumed to be 1640 yds/sec). The excitation matrix $\underset{\sim}{Q}(k)$ can be found as [4]

$$\underset{\sim}{Q}(k) = E\{\underline{W}(k)\cdot\underline{W}(k)^T\}$$

$$\underset{\sim}{Q}(k) = \begin{bmatrix} X_{cpa}\cdot T^2\Sigma_{\dot{\theta}_s}^{\,2} & -R_{cpa}\cdot X_{cpa}\cdot T^2\Sigma_{\dot{\theta}_s}^{\,2} & 0 & A\cdot X_{cpa}\cdot T^2\Sigma_{\dot{\theta}_s}^{\,2} & 0 \\ & \frac{T^4}{4}\Sigma_{\dot{V}_s}^{\,2} + R_{cpa}^{\,2}\cdot T^2\Sigma_{\dot{\theta}_s}^{\,2} & \frac{T^3}{2}\Sigma_{\dot{V}_s}^{\,2} & -A\cdot R_{cpa}\cdot T^2\cdot\Sigma_{\dot{\theta}_s}^{\,2} & 0 \\ & \text{SYMMETRIC} & T^2\Sigma_{\dot{V}_s}^{\,2} & 0 & 0 \\ & & & T^2\cdot\Sigma_{\dot{\theta}_s}^{\,2} & \\ & & & & T^2\cdot\Sigma_{\dot{F}_o}^{\,2} \end{bmatrix} \qquad (6.7)$$

The first two terms involve state related terms.

The state transition matrix is

$$\underset{\sim}{\phi} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (6.8)$$

and the linearized measurement matrix is

$$\underset{\sim}{H}(k) = \frac{\partial \underline{C}}{\partial \underline{X}} \Bigg|_{\underline{X}'(k)}$$

$$= \begin{bmatrix} \dfrac{\partial\theta(k)}{\partial R_{cpa}} & \dfrac{\partial\theta(k)}{\partial X_{cpa}} & \dfrac{\partial\theta(k)}{\partial V_s} & \dfrac{\partial\theta(k)}{\partial \theta_s} & \dfrac{\partial\theta(k)}{\partial F_o} \\[2ex] \dfrac{\partial f(k)}{\partial R_{cpa}} & \dfrac{\partial f(k)}{\partial X_{cpa}} & \dfrac{\partial f(k)}{\partial V_s} & \dfrac{\partial f(k)}{\partial \theta_s} & \dfrac{\partial f(k)}{\partial F_o} \end{bmatrix} \tag{6.9}$$

$$\underset{\sim}{H}(k) = \left[\begin{array}{c|c|c|c|c} -\dfrac{A \cdot X_{cpa}(k)}{R'(k)^2} & \dfrac{A \cdot X_{cpa}(k)}{R'(k)^3} & 0 & 1 & 0 \\[2ex] \hline \dfrac{f'(k)^2 \cdot V_s'(k)}{F_o'(k) \cdot V_p} \cdot \dfrac{X_{cpa}'(k) \cdot R_{cpa}(k)}{R'(k)^3} & -\dfrac{f(k)^2 \cdot V_s'(k)}{F_o'(k) \cdot V_p} \cdot \dfrac{R_{cpa}'(k)}{R'(k)^3} & -\dfrac{f'(k)^2 \cdot X'_{cpa}(k)}{F_o'(k) \cdot V_p} \cdot \dfrac{X'_{cpa}(k)}{R'(k)} & 0 & \dfrac{f'(k)}{F_o'(k)} \end{array}\right] \tag{6.10}$$

where $\underline{X}'(k)$ represents the known track if the analytical equations are used and the predicted states ($\hat{\underline{X}}(k/k-1)$) in the actual extended-Kalman filter. $f'(k)$ is the predicted frequency measurement in the extended-Kalman filter and the

true frequency measurement in the analytical equations.
$R'(k)$ is defined as

$$R'(k) = \sqrt{R'_{cpa}(k)^2 + X'_{cpa}(k)^2} \tag{6.11}$$

and the covariance of measurement noise matrix is

$$\underset{\sim}{R} = \begin{bmatrix} \sigma_\theta^2 & 0 \\ 0 & \sigma_f^2 \end{bmatrix} \tag{6.12}$$

$$= \begin{bmatrix} 7.61543 \times 10^{-3} & 0 \\ 0 & 1.6 \times 10^{-3} \end{bmatrix}$$

The filter was simulated using the analytical equations and the Monte-Carlo simulation with the initial conditions given below. The time between measurements was assumed to be 100 seconds.

$$\hat{\underline{X}}(0/-1) = \begin{bmatrix} 3.15 \times 10^3 \text{ yds.} \\ -2.61 \times 10^3 \text{ yds.} \\ 3.38 \text{ yds/sec} \\ 5.30213 \text{ Rad.} \\ 500.432 \text{ Hz.} \end{bmatrix} \tag{6.13}$$

The $\underset{\sim}{P}(0/-1)$ matrix is

$$\underset{\sim}{P}(0/-1)=\begin{bmatrix} 1.6245\times10^{7} & 0 & 0 & 0 & 0 \\ 0 & 4.057\times10^{6} & 0 & 0 & 0 \\ 0 & 0 & 2.853 & 0 & 0 \\ 0 & 0 & 0 & 0.596 & 0 \\ 0 & 0 & 0 & 0 & 0.684 \end{bmatrix} \tag{6.14}$$

and the true target initial conditions are

$$\underline{X}(0) = \begin{bmatrix} 2992.59 \text{ Yds.} \\ -5295.85 \text{ Yds.} \\ 4.504 \text{ Yds/sec.} \\ 5.06145 \text{ Rad.} \\ 500 \text{ Hz.} \end{bmatrix} \tag{6.15}$$

These initial conditions have been obtained from a method given in [4].)

The analytical equations predict the mean of estimation error to be oscillatory. Monte-Carlo simulation results have indicated that the filter was unstable. The estimation error becomes unbounded as the target passes through CPA. The reason is that the system has two angle measurement equations used depending on the sign of X_{cpa} (see Equation (6.5)). Since the filter has significant estimation error up to CPA, the filter "crosses" the CPA at a different measurement time than the track does. Thus, at a certain instant the filter uses a different angle prediction equation

then the actual measurement equation. This results in a large residual in the estimation equation and the filter begins to diverge. If the filter has estimates with very small error before CPA, the estimator and the target would "cross" the CPA in the same sampling interval. This is a very difficult task because the geometry of the system indicates that any fluctuations in the target heading largely affect the R_{cpa} and X_{cpa}.

In order to force the sign of $\hat{X}_{cpa}(k/k-1)$ to change in the same sampling interval as the track, frequency measurements were tested to determine whether they were "up doppler" or "down doppler", then the sign of the $\hat{X}_{cpa}(k/k-1)$ was reversed accordingly so that X_{cpa} and $\hat{X}_{cpa}(k/k-1)$ have the same sign.

The results of this approach are illustrated in Figures 6.2 through 6.11 for 1000 Monte-Carlo runs shown as the continuous curves. The triangles represent the results obtained from the analytical equations.

Comparing the results of the two methods, one can see that they are completely different (especially the covariances). There are several possible reasons for this, including:

(1) The extended-Kalman filter uses the estimates and predictions for calculation of the $\underset{\sim}{\phi}(k)$, $\underset{\sim}{H}(k)$ and $\underset{\sim}{Q}(k)$ matrices, whereas the true target track was used in the analytical equations. Thus, it is reasonable that the gain schedules will differ largely if the estimates are poor.

(2) Since the linearized measurement matrix $\underset{\sim}{H}(k)$ has been used, the analytical equations do not include the time sharing angle measurement equations; as far as the analytical equations are concerned there is only one angle measurement equation, because the two equations yield the same derivative.

(3) Finally, the analytical equations do not "see" the change that has been made in the filter algorithm to force the filter to cross the CPA in the same measurement interval with the target in the Monte-Carlo simulation.

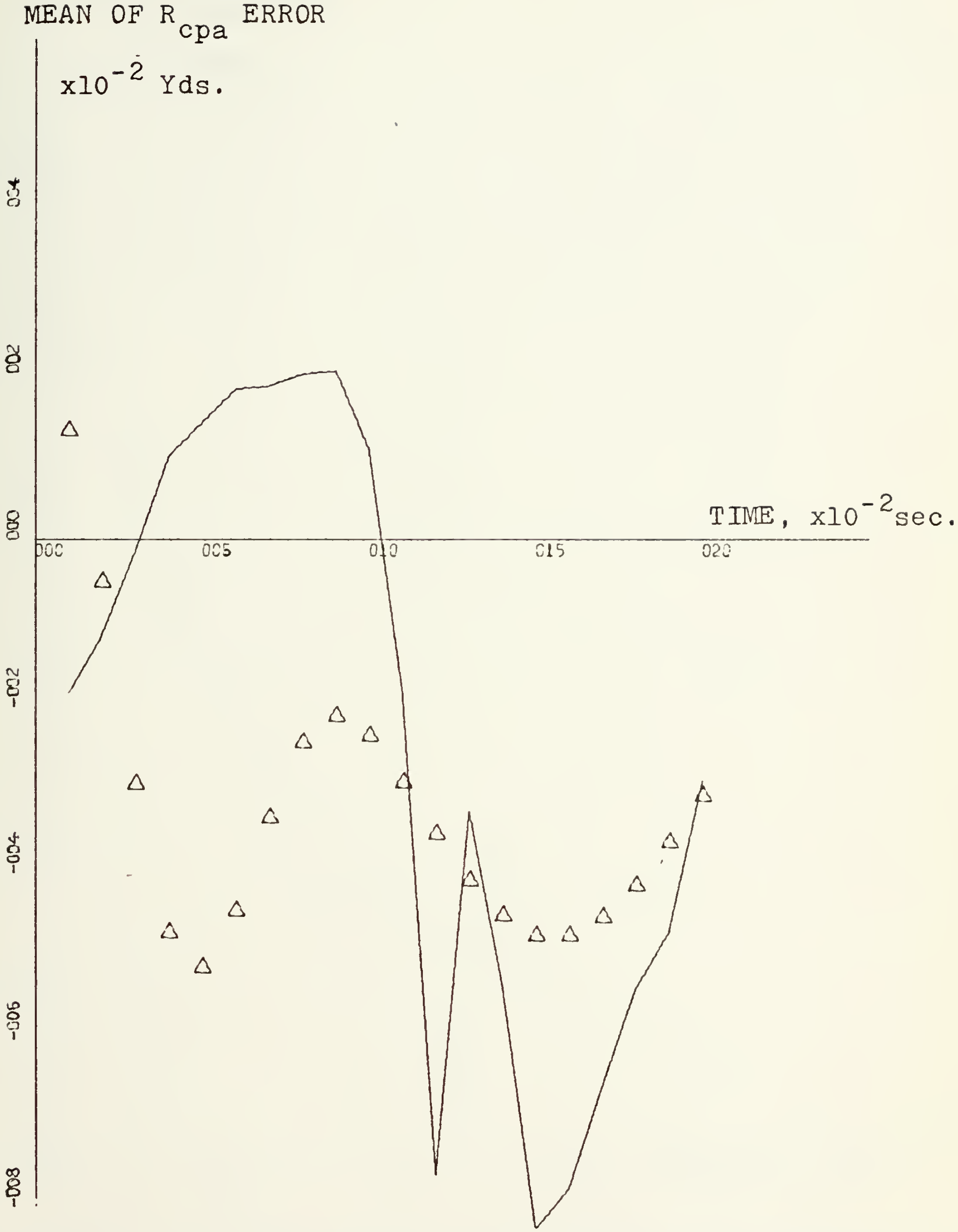

FIG. 6.2. Time history of the mean R_{cpa} estimation error.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=2.00E+02 UNITS INCH.

FIG. 6.3. Time history of the mean X_{cpa} estimation error.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=5.00E+02 UNITS INCH.

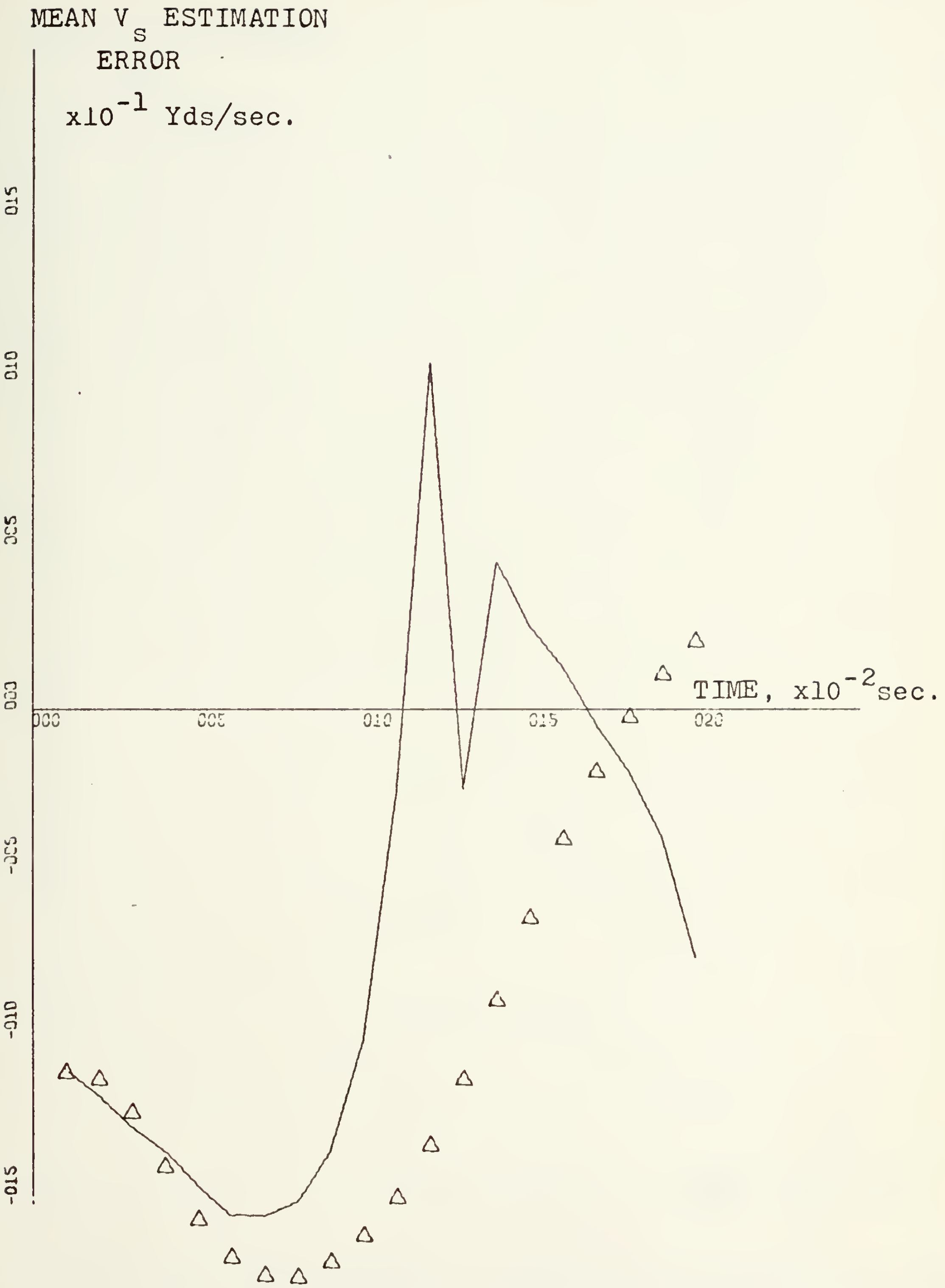

FIG. 6.4. Time history of the mean V_s estimation error.

X-SCALE=5.00E+00 UNITS INCH.

Y-SCALE=5.00E-01 UNITS INCH.

FIG. 6.5. Time history of the mean θ_s estimation error.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=2.00E-01 UNITS INCH.

FIG. 6.6. Time history of the mean F_o estimation error.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=1.00E-01 UNITS INCH.

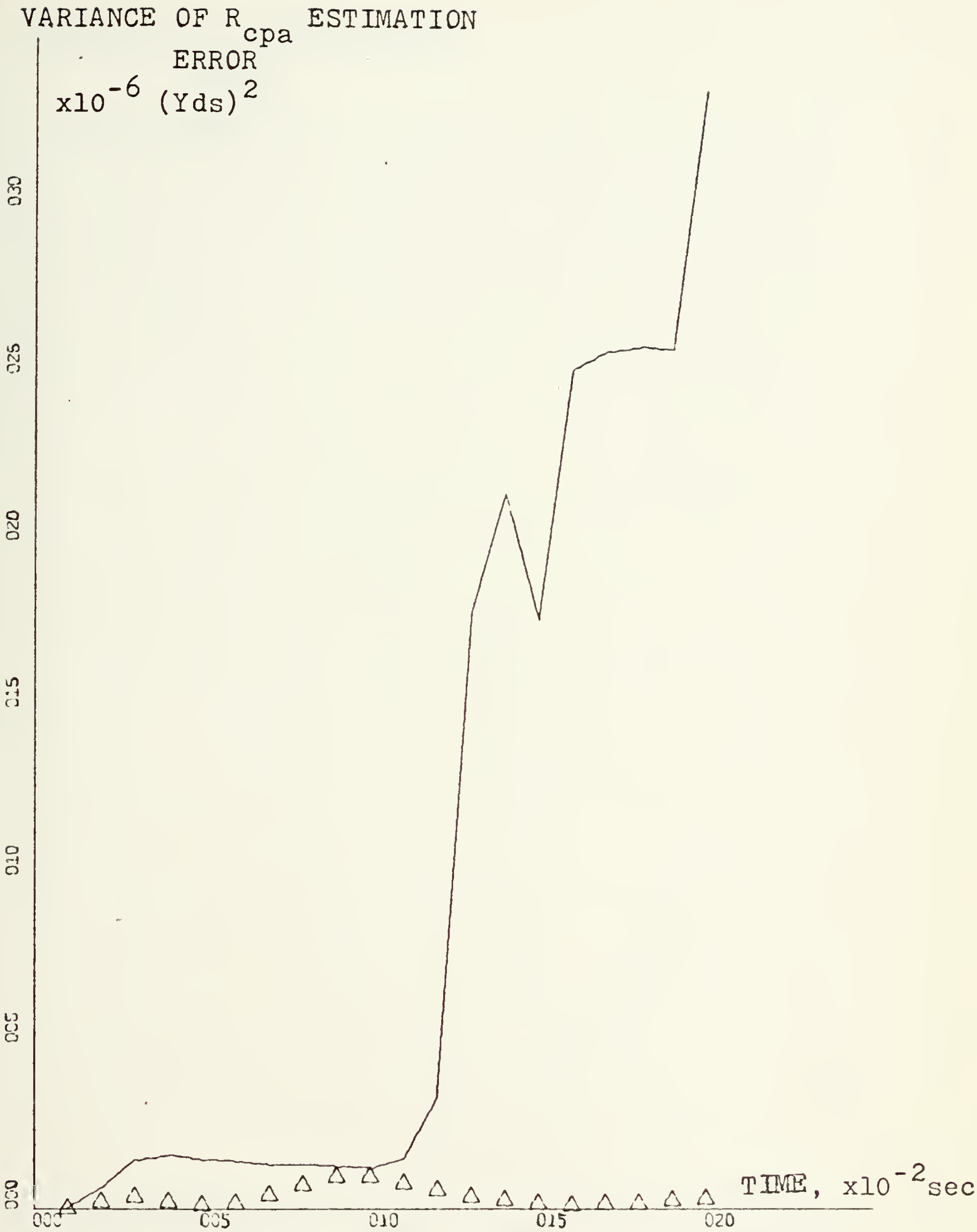

FIG. 6.7. Variance of the mean R_{cpa} estimation error vs. time.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=5.00E+06 UNITS INCH.

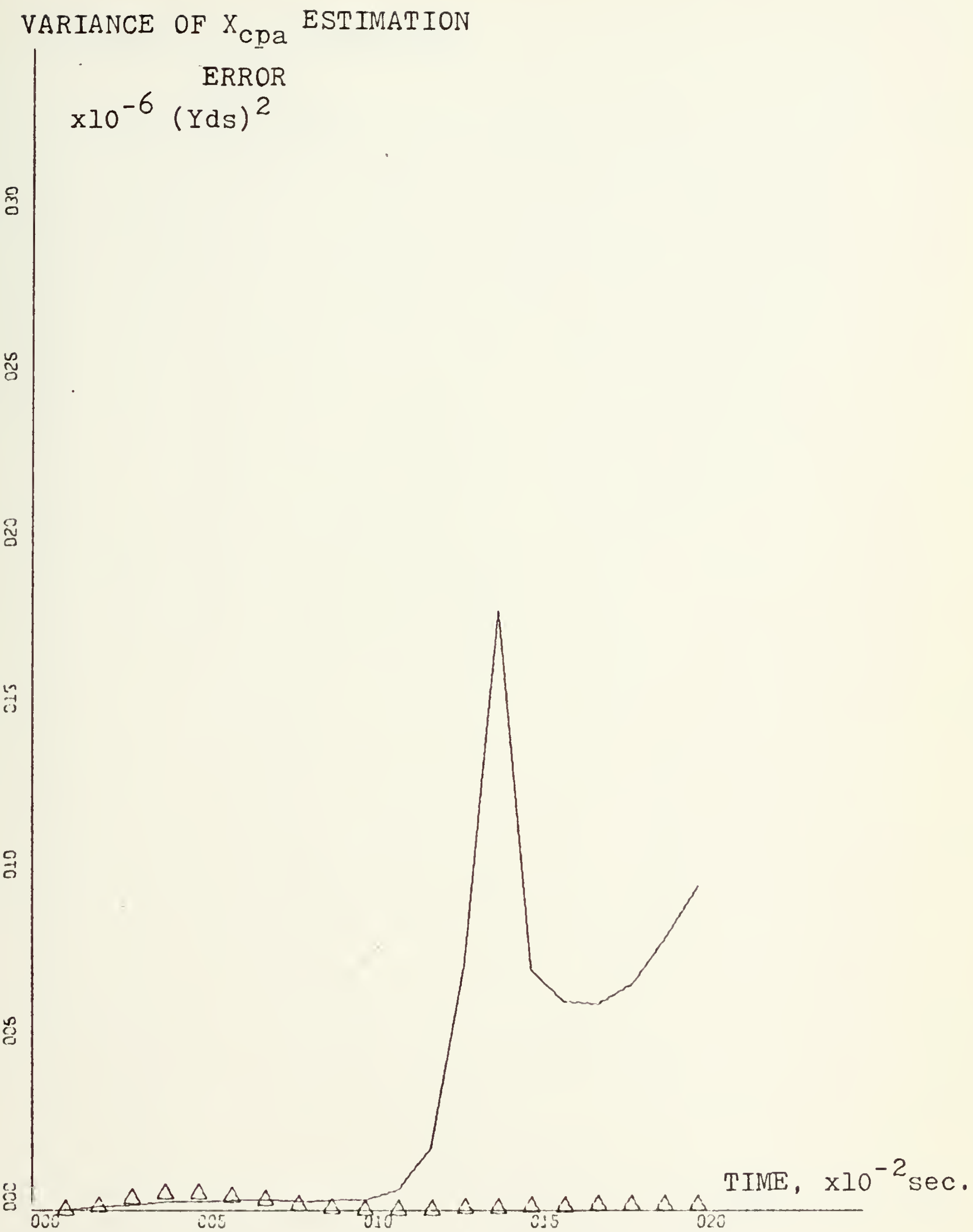

FIG. 6.8. Variance of the mean X_{cpa} estimation error vs. time.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=5.00E+06 UNITS INCH.

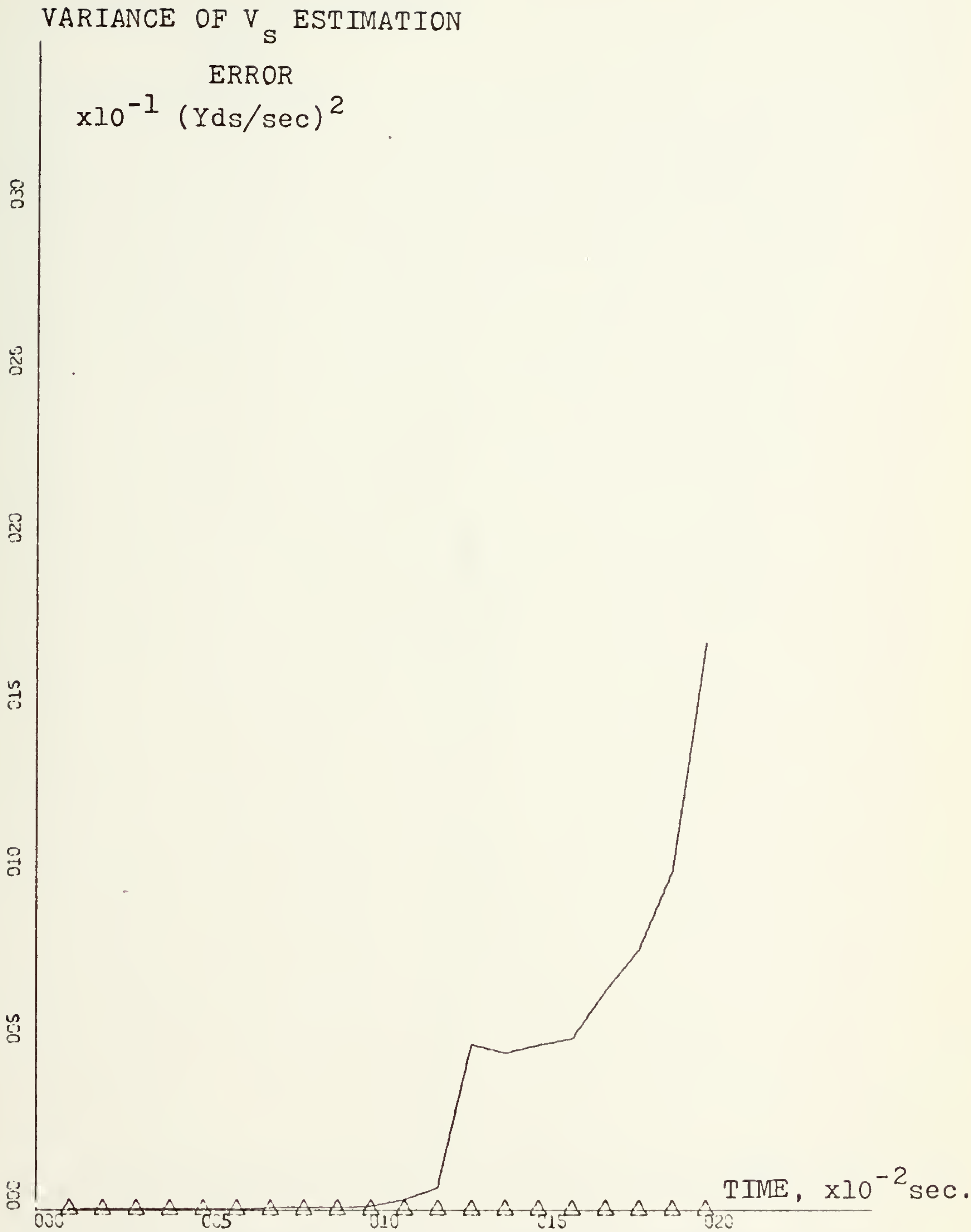

FIG. 6.9. Variance of the mean V_s estimation error vs. time.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=5.00E+01 UNITS INCH.

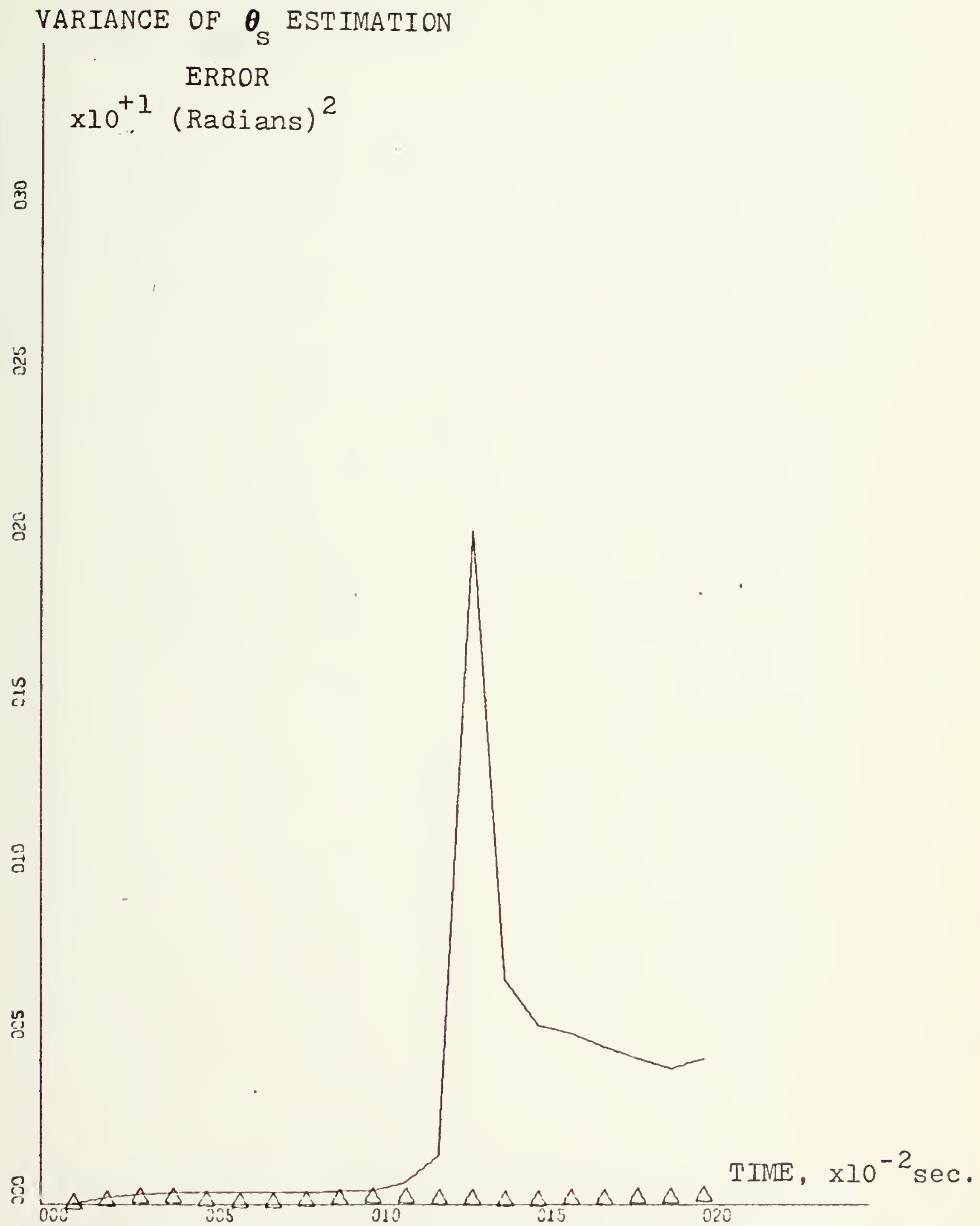

FIG. 6.10. Variance of the mean θ_s estimation error vs. time.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=5.00E-01 UNITS INCH.

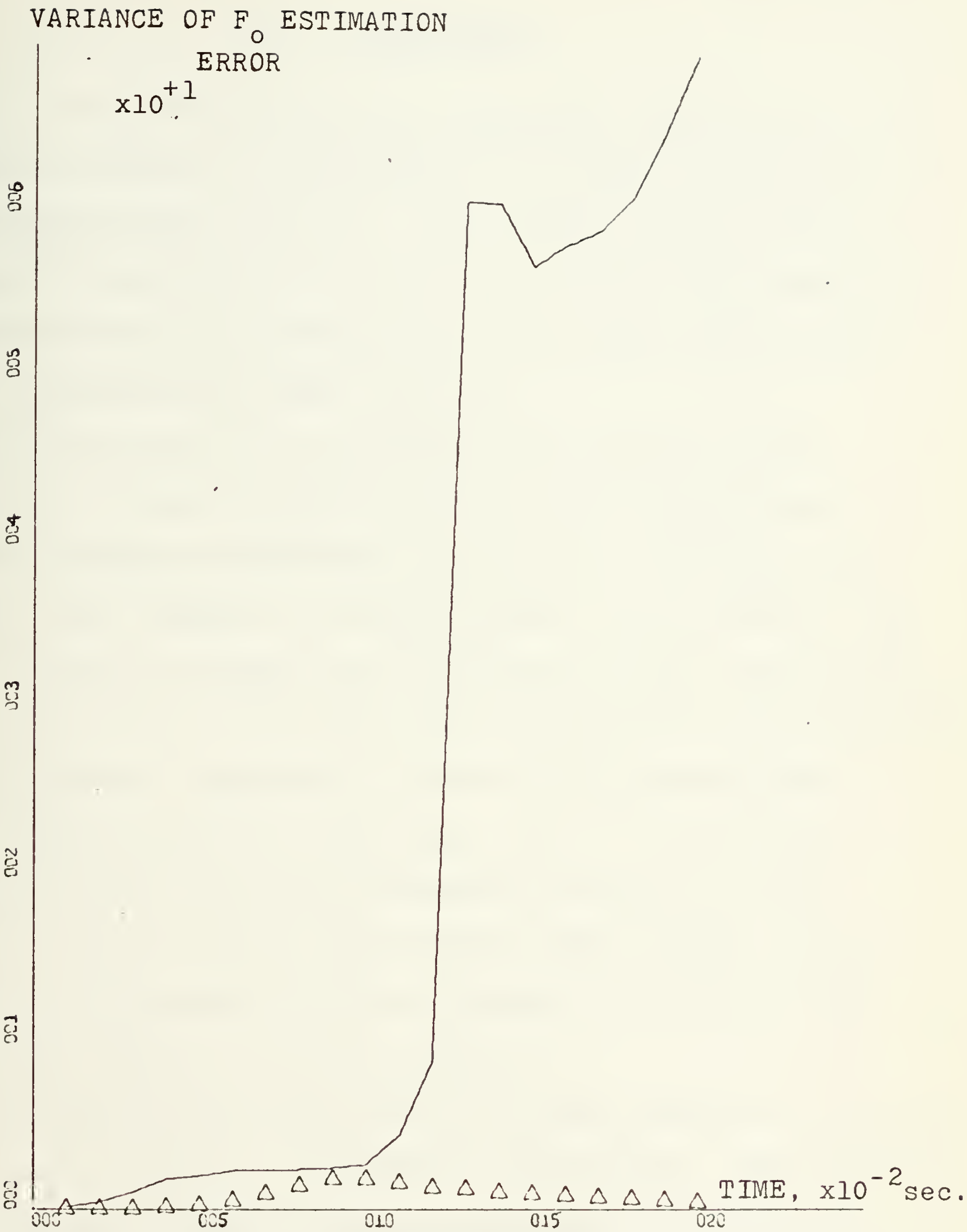

FIG. 6.11. Variance of the mean F_o estimation error vs. time.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=1.00E-01 UNITS INCH.

VII. CONCLUSIONS

It has been shown that the analytical equations can be used to evaluate the performance of a linear filter with a pre-computed gain schedule. The results obtained from the analytical equations are exact and are only closely approximated by Monte-Carlo simulations with large ensemble sizes and correspondingly large CPU times.

For the case of the extended-Kalman filter, it is possible to obtain analytical results which approximate those obtained by Monte-Carlo simulations. How close the results are, depends on the performance of the filter. The main difference between the two methods is that one uses the true track whereas the other uses the estimates and predictions as the nominal trajectory to evaluate the matrices $\underset{\sim}{\phi}(k)$, $\underset{\sim}{H}(k)$ (and $\underset{\sim}{Q}(k)$, $\underset{\sim}{R}(k)$ in some cases, too). If the filter performs poorly and gives diverging estimates, one can expect that the matrices $\underset{\sim}{\phi}(k)$, $\underset{\sim}{H}(k)$, $\underset{\sim}{Q}(k)$ and $\underset{\sim}{R}(k)$ will be completely different in the two methods. This will cause different results in the two methods.

The matrix $\underset{\sim}{Q}(k)$ effects only the gain schedule. If it is a constant matrix or independent of the states, there will be no difference between results obtained by the analytical equations and Monte-Carlo simulation. The matrix $\underset{\sim}{R}(k)$ has two kinds of effects. One is similar to that which $\underset{\sim}{Q}(k)$ has, the other one is that the R(k) matrix represents the covariance

of measurement noise, and in Chapter V, it has been shown that the "larger" $\underset{\sim}{R}(k)$ is the, larger the difference between results becomes. From the results given in Chapter V, it is observed that the initial filter states effect the results too. Depending on how close the initial filter states are to the true initial states, the time at which the filter converges will change.

In summary, one can use the analytical equations to approximately predict the performance of an extended-Kalman filter. The accuracy of prediction depends on the nature of the problem and the filter.

APPENDIX A

COMPUTER PROGRAMS

A. PROGRAM "EVAL"

The Program "EVAL" is a computer algorithm which solves the analytical equations derived in Chapter II. The algorithm has been written for the special case of the analytical equations which is developed for the MK. 86 Fire Control System as discussed in Reference [1]. The estimation equation has been assumed as

$$\underline{\hat{X}}(k/k) = \underline{\hat{X}}(k/k-1) + \underset{\sim}{G}(k) \Big[\underset{\sim}{M}_o(k)\ \underline{Z}(k) + \underset{\sim}{M}_1(k)\ \underline{Z}(k-1) + \underset{\sim}{M}_2(k)\ \underline{Z}(k-2) - \underset{\sim}{C}\ \underline{\hat{X}}(k/k-1)\Big] \qquad (A.1)$$

and the measurement equation as

$$\underline{Z}(k) = \underset{\sim}{H}\ \underline{X}(k) + \underline{V}(k) \qquad (A.2)$$

where $\underset{\sim}{H}$ is the true measurement matrix $\underset{\sim}{C}$ is the measurement matrix with synthetic measurements. The matrices $\underset{\sim}{M}_o(k)$, $\underset{\sim}{M}_1(k)$, $\underset{\sim}{M}_2(k)$ are measurement weighting matrices which have been used in the MK. 86 Fire Control System.

In the development of this research these matrices are defined as

$$\underset{\sim}{M}_o(k) = \underset{\sim}{I} \qquad \text{(qxq Identity matrix)} \qquad (A.3)$$

$$\underset{\sim}{M}_1(k) = \underset{\sim}{M}_2(k) = \underset{\sim}{0} \qquad (A.4)$$

$$\underset{\sim}{C} = \underset{\sim}{H} \qquad (A.5)$$

Then Equation (A.1) reduces to the simpler form

$$\hat{\underline{X}}(k/k) = \hat{\underline{X}}(k/k-1) + \underset{\sim}{G}(k)\left[\ \underline{Z}(k) - \underset{\sim}{C}\ \hat{\underline{X}}(k/k-1)\right] \qquad (A.6)$$

With the assumptions given above and starting with the definition of the estimation error, i.e.

$$\tilde{\underline{X}}(k) = \hat{\underline{X}}(k/k) - \underline{X}(k) \qquad (A.7)$$

(where $\tilde{\underline{X}}(k)$ is the true estimation error vector) the following analytical results have been derived in [1]:

1. Covariance of estimation error

$$\begin{aligned}\tilde{\underset{\sim}{P}}(k/k) = {} & \underset{\sim}{S}_3(k)\ \tilde{\underset{\sim}{P}}(k-1/k-1)\ \underset{\sim}{S}_3(k)^T + \underset{\sim}{S}_o(k)\ \underset{\sim}{R}(k)\ \underset{\sim}{S}_o(k)^T \\ & +\underset{\sim}{S}_1(k)\ \underset{\sim}{R}(k-1)\ \underset{\sim}{S}_1(k)^T + \underset{\sim}{S}_2(k)\ \underset{\sim}{R}(k-2)\ \underset{\sim}{S}_2(k)^T \\ & +\underset{\sim}{A}_1(k) + \underset{\sim}{A}_1(k)^T + \underset{\sim}{A}_2(k) + \underset{\sim}{A}_2(k)^T \qquad (A.8)\end{aligned}$$

2. Mean of estimation error

$$\tilde{\underline{\mu}}\ (k) = \underset{\sim}{S}_3(k)\ \tilde{\underline{\mu}}\ (k-1) + \underset{\sim}{D}(k) + \underset{\sim}{S}_3(k)\ \underline{X}(k-1) \qquad (A.9)$$

3. Initializing values

$$\tilde{\underset{\sim}{P}}(0/0) = \underset{\sim}{S}_o(0)\ R(0)\ \underset{\sim}{S}_o(0)^T \qquad (A.10)$$

$$\tilde{\underline{\mu}}\ (0) = \left[\underset{\sim}{I} - \underset{\sim}{G}(0)\ \underset{\sim}{C}\right]\ \hat{\underline{X}}(0/-1) - \left[\underset{\sim}{I} - \underset{\sim}{G}(0)\ \underset{\sim}{M}_o(0)\underset{\sim}{H}\right]\underline{X}(0) \qquad (A.11)$$

4. Mean of N-step prediction error

$$\tilde{\underline{\mu}}_p\ (N+k/k) = \underset{\sim}{\phi}^N\ \tilde{\underline{\mu}}(k) + \underset{\sim}{\phi}^N\ \underline{X}(k) - \underline{X}(N+k) \qquad (A.12)$$

5. Covariance of N-step prediction error

$$\tilde{\underset{\sim}{P}}(N+k/k) + \underset{\sim}{\phi}^N \tilde{\underset{\sim}{P}}(k/k) \left[\underset{\sim}{\phi}^N \right]^T \qquad (A.13)$$

where

$$\underset{\sim}{S}_o(k) = \underset{\sim}{G}(k) \underset{\sim}{M}_o(k)$$

$$\underset{\sim}{S}_1(k) = \underset{\sim}{G}(k) \underset{\sim}{M}_1(k)$$

$$\underset{\sim}{S}_2(k) = \underset{\sim}{G}(k) \underset{\sim}{M}_2(k)$$

$$\underset{\sim}{S}_3(k) = \left[\underset{\sim}{I} - \underset{\sim}{G}(k) \underset{\sim}{C} \right] \underset{\sim}{\phi}$$

$$\underset{\sim}{A}_1(k) = \underset{\sim}{S}_3(k) \underset{\sim}{S}_o(k-1) \underset{\sim}{R}(k-1) \underset{\sim}{S}_1(k)^T$$

$$\underset{\sim}{A}_2(k) = \underset{\sim}{S}_3(k) \left[\underset{\sim}{S}_3(k-1) \underset{\sim}{S}_o(k-2) + \underset{\sim}{S}_1(k-1) \right] \underset{\sim}{R}(k-1) \underset{\sim}{S}_2(k)^T$$

$$\underset{\sim}{D}(k) = \left[\underset{\sim}{S}_o(k) \underset{\sim}{H} - \underset{\sim}{I} \right] \underline{X}(k) + \underset{\sim}{S}_1(k) \underset{\sim}{H} \underline{X}(k-1) + \underset{\sim}{S}_2(k) \underset{\sim}{H} \underline{X}(k-1)$$

With the assumptions given by Equations (A.3) - (A.5), Equations (A.8) through (A.13) reduce to the form of equations derived in Chapter II.

The computer program given in [1] has been used after making some small changes and adding three subroutines. The program listing is given at the end of this Appendix. In order to use the program, one must provide subroutine AK which computes the linearized state transition matrix $\phi(k)$, subroutine H(k) which calculates the matrix HKK and subroutine UK which calculates the vector $\underline{U}'(k-1)$. If the matrices $\underset{\sim}{Q}(k)$ and $\underset{\sim}{R}(k)$ are required to be calculated on-line,

then subroutines QON and RON must also be provided. It is also necessary to give the proper values to certain flags for required operation.

B. MONTE-CARLO SIMULATION

A listing of the program used for Monte-Carlo simulation is also given at the end of this Appendix. The program has been set up for use with both linear and extended-Kalman filters. One can simulate a linear or extended-Kalman filter by selecting the proper flags and supplying the proper subroutines. For extended-Kalman filter simulation, one must provide subroutine MEAS for simulation of measurements, subroutine TRACK for generating the target track (if not read in), subroutine HKK for calculation of $H(k)$, subroutine AK for calculation of $\phi(k)$, subroutine CK for calculation of $C(\underline{X}(k/k-1))$, subroutine XPRED for calculation of $\underline{X}(k+1/k) = \underline{f}(\underline{X}(k/k))$, and subroutines QON and RON for $Q(k)$ and $R(k)$ if they are not read in.

```
C        EVAL
C   THIS PROGRAM SOLVES THE EQUATIONS OF CH. 4 TO YIELD MEAN AND       EVAL0000
C   COVARIANCE OF ESTIMATION ERROR FOR A SPECIFIED TARGET              EVAL0001
C   TRAJECTORY. THE ARRAYS ARE SET UP TO  ACCOMODATE A MAX             EVAL0002
C   OF 4 STATE VARIABLES.                                              EVAL0003
C                                                                      EVAL0004
C                                                                      EVAL0005
C   N= NO. OF STATE VARIABLES                                          EVAL0006
C   NA= NO. OF ACTUAL MEASUREMENTS                                     EVAL0007
C   L= NO. OF SYNTHETIC MEASUREMENTS + NO. OF ACTUAL MEASUREMENTS      EVAL0008
C   NSAM= NO. OF SAMPLES                                               EVAL0009
C   INOISE IS AN INDICATOR. IF INOISE.EQ.0 THE MEASUREMENT             EVAL0010
C   NOISE COVARIANCE MATRIX R IS A CONSTANT MATRIX WHICH IS            EVAL0011
C   READ IN, OTHERWISE, R IS CALCULATED BY SUBROUTINE RNOISE           EVAL0012
C     IG=0 -- GAINS COMPUTED ONLINE BY CALLING "GAIN". OTHERWISE, GAINS EVAL0013
C             READ IN.                                                 EVAL0014
C     ITRK=1 -- TRACK COMPUTED BY CALLING "TRACK". OTHERWISE, TRACK    EVAL0015
C               READ IN.                                               EVAL0016
C     ND IS A CONSTANT THAT IS READ IN. IT MUST EQUAL THE DIMENSION M  EVAL0017
C     OF THE MXM ARRAY TMP1.                                           EVAL0018
C                                                                      EVAL0019
      IMPLICIT REAL*8 (A-H,O-Z)                                        EVAL0020
      REAL*8 MOK,M1K,M2K,MUK,MUKS                                      EVAL0021
      COMMON GKS(9,9,60),PKS(9,9,60),XKS(9,60),MUKS(9,60),SOK(9,9),
     1S1K(9,9),S2K(9,9),S3K(9,9),PHI(9,9),C(9,9),H(9,9),EID(9,9),
     2A1K(9,9),A2K(9,9),GK(9,9),MOK(9,9),M1K(9,9),M2K(9,9),SOKM1(9,9),
     3S1KM1(9,9),RK(9,9),SOKM2(9,9),RKM1(9,9),RKM2(9,9),S3KM1(9,9),
     4TMP1(9,9),TMP2(9,9),PK(9,9),XK(9),VTMP1(9),XHIT(9),VTMP2(9),
     5CON(9),VTMP3(9),MUK(9),XKM1(9),XKM2(9),DK(9),N,L,NA,K,ND,NSAM,
     6IP,IP1,IH,IQ,PHK(9,9,60),HK(9,9,60),QK(9,9,60),PMUKS1(9,60),
     7PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)
      DIMENSION XP(80),YP(80)
C                                                                      EVAL0028
C   PHI= NXN STATE TRANSITION MATRIX                                   EVAL0029
C   C= LXN SYNTHETIC MEASUREMENT MATRIX                                EVAL0030
C   H= (NA)XN ACTUAL MEASUREMENT MATRIX                                EVAL0031
C   MOK,M1K,M2K ARE LX(NA) MATRICES USED TO FORM THE LINEAR            EVAL0032
C   COMBINATIONS OF THE MEASUREMENTS                                   EVAL0033
C   EID= NXN IDENTITY MATRIX                                           EVAL0034
C   GKS= NXLX(NSAM) ARRAY OF STORED FILTER GAINS                       EVAL0035
C   XKS= NX(NSAM) ARRAY OF STORED TRAJECTORY VALUES                    EVAL0036
C   RK= (NA)X(NA) COVARIANCE OF MEASUREMENT NOISE MATRIX               EVAL0037
C   MUKS= NX(NSAM) ARRAY OF MEAN OF EST. ERROR HISTORY                 EVAL0038
C   PKS= NXNX(NSAM) ARRAY OF COVARIANCE OF EST. ERROR HISTORY          EVAL0039
C     SOK(N,NA),S1K(N,NA),S2K(N,NA),S3K(N,N),DK(N),A1K(N,N) ,A2K(N,N)  EVAL0040
C     ARE AS DEFINED IN CH. 4                                          EVAL0041
C                                                                      EVAL0042
C   IP=0  PH IS GOING TO BE COMPUTED ON LINE
```

```
C
C     IH=0  H MATRIX WILL BE COMPUTED ONLINE
C     IQ=0  QMATRIX WILL BE COMPUTED ONLINE
C     IP1=0   TWO TRACKS ARE GOING TO BE USED  WITH GIVEN PROB
C                                                                       EVAL0043
      READ (5,55) N,L,NA,INOISE,IG,ITRK,NSAM,ND                         EVAL0044
      READ(5,55) IP,IP1,IH,IQ
      IF(IP1.EQ.0) READ (5,56) P
      IP2=1
C     FORM NXN IDENTITY MATRIX EID                                      EVAL0045
C                                                                       EVAL0046
      DO 2 I=1,N                                                        EVAL0047
C                                                                       EVAL0048
      DO 2 J=1,N                                                        EVAL0049
      IF (I.EQ.J) GO TO 1                                               EVAL0050
      EID(I,J) = 0.D0                                                   EVAL0051
      GO TO 2                                                           EVAL0052
    1 EID(I,J) = 1.D0                                                   EVAL0053
    2 CONTINUE                                                          EVAL0054
C                                                                       EVAL0055
      READ (5,56) (XHIT(I),I=1,N)                                       EVAL0056
C                                                                       EVAL0057
      DO 3 I=1,L                                                        EVAL0058
    3 READ (5,56) (M0K(I,J),J=1,NA)                                     EVAL0059
C                                                                       EVAL0060
C                                                                       EVAL0061
      DO 4 I=1,L                                                        EVAL0062
    4 READ (5,56) (M1K(I,J),J=1,NA)                                     EVAL0063
C                                                                       EVAL0064
C                                                                       EVAL0065
      DO 5 I=1,L                                                        EVAL0066
    5 READ (5,56) (M2K(I,J),J=1,NA)                                     EVAL0067
C                                                                       EVAL0068
      IF (IP.EQ.0) GO TO 6
C                                                                       EVAL0069
      DO 6 I=1,N                                                        EVAL0070
      READ (5,56) (PHI(I,J),J=1,N)
    6 CONTINUE
C                                                                       EVAL0072
      IF(IH.EQ.0) GO TO 8
C                                                                       EVAL0073
      DO 7 I=1,L                                                        EVAL0074
    7 READ (5,56) (C(I,J),J=1,N)                                        EVAL0075
C                                                                       EVAL0076
C                                                                       EVAL0077
      DO 8 I=1,NA                                                       EVAL0078
      READ (5,56) (H(I,J),J=1,N)
    8 CONTINUE
```

```
C
   85 WRITE (6,57)
      WRITE (6,58) N,L,NA,INOISE,NSAM
      IF (IG.EQ.0) GO TO 10
C
      DO 9 K=1,NSAM
C
      DO 9 I=1,N
    9 READ (5,59) (GKS(I,J,K),J=1,L)
C
      GO TO 16
C
   10 DO 11 I=1,N
   11 READ (5,56) (PK(I,J),J=1,N)
C
C     P(0/-1) READ IN TO ARRAY PK FOR GAIN CALCULATION.
      WRITE (6,83)
C
      DO 84 I=1,N
   84 WRITE (6,63) (PK(I,J),J=1,N)
C
      IF(IQ.EQ.0) GO TO 13
C
      DO 12 I=1,N
   12 READ (5,56) (RKM2(I,J),J=1,N)
C
      WRITE (6,60)
C
      DO 13 I=1,N
      WRITE (6,63) (RKM2(I,J),J=1,N)
   13 CONTINUE
C
C     Q READ IN TO ARRAY RKM2 FOR GAIN CALCULATION
C
      DO 14 I=1,NA
   14 READ (5,56) (RKM1(I,J),J=1,NA)
C
      WRITE (6,61)
C
      DO 15 I=1,NA
   15 WRITE (6,63) (RKM1(I,J),J=1,NA)
C
C     R USED FOR GAIN CALCULATION READ IN AND STORED IN RKM1. NOTE THAT
C     R IS ASSUMED TO BE NA X NA.
   16 IF (ITRK.EQ.1) GO TO 18
C
      DO 17 K=1,NSAM
   17 READ (5,59) (XKS(I,K),I=1,N)
```

```
C                                                                       EVAL0122
      GO TO 19                                                          EVAL0123
   18 CALL TRACK                                                        EVAL0124
C     THE ARRAY CON IS FOR STORAGE OF CONSTANTS WHICH MAY BE NEEDED     EVAL0125
C     IN SUBROUTINES.                                                   EVAL0126
   19 READ (5,56) (CON(I),I=1,8)                                        EVAL0127
      IF(IP.EQ.0) CALL AK
      IF(IH.EQ.0) CALL HKK
      IF(IQ.EQ.0) CALL QON
      IF (IG.EQ.0) CALL GAIN
      IF (INOISE.NE.0) GO TO 22                                         EVAL0128
C                                                                       EVAL0129
      DO 20 I=1,NA                                                      EVAL0130
   20 READ (5,56) (RK(I,J),J=1,NA)                                      EVAL0131
C                                                                       EVAL0132
C                                                                       EVAL0133
      DO 21 I=1,NA                                                      EVAL0134
C                                                                       EVAL0135
      DO 21 J=1,NA                                                      EVAL0136
      RKM1(I,J) = RK(I,J)                                               EVAL0137
   21 RKM2(I,J) = RK(I,J)                                               EVAL0138
C                                                                       EVAL0139
   22 CONTINUE                                                          EVAL0140
C     WRITE OUT PARTS OF INPUT DATA                                     EVAL0141
      WRITE (6,62)                                                      EVAL0142
C                                                                       EVAL0143
      DO 23 I=1,N                                                       EVAL0144
   23 WRITE (6,63) (GKS(I,J,1),J=1,L)                                   EVAL0145
C                                                                       EVAL0146
C                                                                       EVAL0147
      DO 24 I=1,N                                                       EVAL0148
   24 WRITE (6,63) (GKS(I,J,NSAM),J=1,L)                                EVAL0149
C                                                                       EVAL0150
      IF (IG.NE.0) GO TO 27                                             EVAL0151
      WRITE (6,64)                                                      EV1
C                                                                       EVAL0153
      DO 25 I=1,N                                                       EVAL0154
   25 WRITE (6,63) (PKS(I,J,1),J=1,N)                                   EVAL0155
C                                                                       EVAL0156
C                                                                       EVAL0157
      DO 26 I=1,N                                                       EVAL0158
   26 WRITE (6,63) (PKS(I,J,NSAM),J=1,N)                                EVAL0159
C                                                                       EVAL0160
   27 WRITE (6,65)                                                      EVAL0161
      WRITE (6,63) (XKS(I,1),I=1,N)                                     EVAL0162
      WRITE (6,63) (XKS(I,NSAM),I=1,N)                                  EVAL0163
      WRITE (6,66)                                                      EVAL0164
      WRITE (6,63) (XHIT(I),I=1,N)                                      EVAL0165
```

```
      IF(IP.EQ.0) GO TO 28
      WRITE (6,67)
C
      DO 28 I=1,N
      WRITE (6,63) (PHI(I,J),J=1,N)
   28 CONTINUE
      IF(IH.EQ.0) GO TO 86
C
      WRITE (6,68)
C
      DO 29 I=1,NA
   29 WRITE (6,63) (H(I,J),J=1,N)
C
      WRITE (6,69)
C
      DO 30 I=1,L
   30 WRITE (6,63) (C(I,J),J=1,N)
C
   86 WRITE (6,70)
C
      DO 31 I=1,L
   31 WRITE (6,63) (M0K(I,J),J=1,NA)
C
      WRITE (6,71)
C
      DO 32 I=1,L
   32 WRITE (6,63) (M1K(I,J),J=1,NA)
C
      WRITE (6,72)
C
      DO 33 I=1,L
   33 WRITE (6,63) (M2K(I,J),J=1,NA)
C
      IF (INOISE.NE.0) GO TO 35
      WRITE (6,73)
C
      DO 34 I=1,NA
   34 WRITE (6,63) (RK(I,J),J=1,NA)
C
   35 CONTINUE
      WRITE (6,74)
C      COMPUTATION OF MEANS AND COVARIANCES OF ESTIMATION ERROR
C     BEGINS HERE. THE FIRST TWO SAMPLES REQUIRE SOME ATYPICAL
C     SETTINGS FOR SOME OF THE MATRICES INVOLVED AND SO ARE
C     DONE OUTSIDE OF THE MAIN LOOP.
C
C
C     COMPUTE S0(0)=S0K
```

```
      K = 1                                                             EVAL0211
      CALL LOADG                                                        EVAL0212
      CALL PROD (GK,MOK,N,L,NA,SOK)                                     EVAL0213
C     SET S1(0),S2(0),A1(0),A2(0),S3(0) TO ZERO SO THAT SUBROUTINE      EVAL0214
C     COVE CAN BE USED TO COMPUTE P(0) AS GIVEN BY EQ. (4.38)           EVAL0215
      IF (INOISE.NE.0) CALL RNOISE                                      EVAL0216
C                                                                       EVAL0217
      DO 36 I=1,N                                                       EVAL0218
C                                                                       EVAL0219
      DO 36 J=1,NA                                                      EVAL0220
      S1K(I,J) = 0.D0                                                   EVAL0221
   36 S2K(I,J) = 0.D0                                                   EVAL0222
C                                                                       EVAL0223
C                                                                       EVAL0224
      DO 37 I=1,N                                                       EVAL0225
C                                                                       EVAL0226
      DO 37 J=1,N                                                       EVAL0227
      A1K(I,J) = 0.D0                                                   EVAL0228
      A2K(I,J) = 0.D0                                                   EVAL0229
   37 S3K(I,J) = 0.D0                                                   EVAL0230
C                                                                       EVAL0231
      CALL COVE                                                         EVAL0232
C     P(0) HAS BEEN COMPUTED                                            EVAL0233
C     COMPUTE MU(0) -- SUBROUTINE MEANE NOT USED THIS TIME ONLY         EVAL0234
C                                                                       EVAL0235
      DO 38 I=1,N                                                       EVAL0236
      XK(I) = XKS(I,K)                                                  EVAL0237
   38 VTMP1(I) = XHIT(I)-XK(I)                                          EVAL0238
C
      IF(IH.NE.0) GO TO 88
C
      DO 88 I=1,NA
      DO 88 J=1,N
      H(I,J)=HK(I,J,K)
      C(I,J)=HK(I,J,K)
   88 CONTINUE
C                                                                       EVAL0239
      CALL VPROD (C,XHIT,L,N,VTMP2)                                     EVAL0240
      CALL VPROD (H,XK,NA,N,VTMP3)                                      EVAL0241
      CALL VPROD (MOK,VTMP3,L,NA,VTMP3)                                 EVAL0242
      CALL VSUB (VTMP3,VTMP2,L,VTMP2)                                   EVAL0243
      CALL VPROD (GK,VTMP2,N,L,MUK)                                     EVAL0244
      CALL VADD (MUK,VTMP1,N,MUK)                                       EVAL0245
C     COMPUTATION OF MU(0) COMPLETE                                     EVAL0246
C     STORE P(0), MU(0)                                                 EVAL0247
      CALL STORE                                                        EVAL0248
C     SET UP FOR NEXT CYCLE                                             EVAL0249
      K = 2                                                             EVAL0250
```

```
C     REPLACE XKM1 BY XK, REPLACE SOKM1 BY SOK, RESET XK
C
      DO 39 I=1,N
      XKM1(I) = XK(I)
      XK(I) = XKS(I,K)
C
      DO 39 J=1,NA
   39 SOKM1(I,J) = SOK(I,J)
C
      IF (INOISE.EQ.0) GO TO 41
C     IF INOISE.NE.0 COMPUTE NEW VALUE OF R(K)
C
      DO 40 I=1,NA
C
      DO 40 J=1,NA
   40 RKM1(I,J) = RK(I,J)
C
      CALL RNOISE
   41 CONTINUE
C     CALCULATE SO(1)=SOK
      CALL LOADG
      CALL PROD (GK,MOK,N,L,NA,SOK)
C     CALCULATE S1(1)=S1K
      CALL PROD (GK,M1K,N,L,NA,S1K)
C     SET S2(1)=0
C
      DO 42 I=1,N
C
      DO 42 J=1,NA
   42 S2K(I,J) = 0.D0
C
C     CALCULATE S3(1)=S3K
      CALL S3
C     CALCULATE A1(1)=A1K
      CALL A1
C     SET A2(1)=0 BECAUSE S2K=0
C
      DO 43 I=1,N
C
      DO 43 J=1,N
   43 A2K(I,J) = 0.D0
C
C     NOW READY TO COMPUTE P(1)
C     WRITE(6,73) (SOK(I,1),I=1,N)
      CALL COVE
C     SET UP TO COMPUTE MU(1)=MUK FROM  EQ. (4.34)
C     SET XKM2 = 0
C
```

```
      DO 44 I=1,N
   44 XKM2(I) = 0.D0
C
      CALL D
      CALL MEANE
C     WRITE(6,73) (MUK(I),I=1,N)
C     STORE P(1) AND MU(1)
      CALL STORE
C     BEGIN GENERAL DO LOOP (K=3,NSAMPLES) HERE
C
C
C
      DO 49 K=3,NSAM
C     SHIFT LOCATIONS OF MATRICES TO PREPARE FOR NEXT CYCLE
C
      DO 45 I=1,N
      XKM2(I) = XKS(I,K-2)
      XKM1(I) = XKS(I,K-1)
      XK(I) = XKS(I,K)
C
      DO 45 J=1,NA
      SOKM2(I,J) = SOKM1(I,J)
      SOKM1(I,J) = SOK(I,J)
   45 S1KM1(I,J) = S1K(I,J)
C
C
      DO 46 I=1,N
C
      DO 46 J=1,N
   46 S3KM1(I,J) = S3K(I,J)
C
      IF (INOISE.EQ.0) GO TO 48
C     CCMPUTE R(K) IF INOISE.NE.0
C     FIRST SHIFT THE MATRICES RKM1,RK TO RKM2,RKM1,RESP
C
      DO 47 I=1,NA
C
      DO 47 J=1,NA
      RKM2(I,J) = RKM1(I,J)
   47 RKM1(I,J) = RK(I,J)
C
      CALL RNOISE
   48 CONTINUE
      CALL LOADG
C     COMPUTE SO(K)=SOK FROM EQ. (4.33)
      CALL PROD (GK,MOK,N,L,NA,SOK)
C     COMPUTE S1(K)=S1K FROM EQ. (4.33)
      CALL PROD (GK,M1K,N,L,NA,S1K)
```

```
C         COMPUTE S2(K)=S2K FROM EQ. (4.33)                          EVAL0347
          CALL PROD (GK,M2K,N,L,NA,S2K)                              EVAL0348
C         COMPUTE S3(K)=S3K FROM EQ. (4.33)                          EVAL0349
          CALL S3                                                    EVAL0350
C          COMPUTE A1(K)=A1K FROM EQ. (4.33)                         EVAL0351
          CALL A1                                                    EVAL0352
C         COMPUTE A2(K)=A2K FROM EQ. (4.33)                          EVAL0353
          CALL A2                                                    EVAL0354
C         COMPUTE COVARIANCE OF ESTIMATION ERROR                     EVAL0355
C         AT TIME K, P(K)=PK                                         EVAL0356
          CALL COVE                                                  EVAL0357
C         COMPUTE D(K)=DK AND MU(K)=MUK                              EVAL0358
          CALL D                                                     EVAL0359
          CALL MEANE                                                 EVAL0360
C         STORE P(K) AND MU(K)                                       EVAL0361
          CALL STORE                                                 EVAL0362
       49 CONTINUE                                                   EVAL0363
          IF(IP1.NE.0) GO TO 93
          IF(IP2.EQ.2) GO TO 91
          IP2=2
C
          DO 90 K=1,NSAM
C
          DO 90 I=1,N
          PMUKS1(I,K)=MUKS(I,K)
C
          DO 90 J=1,N
          PKS1(I,J,K)=PKS(I,J,K)
       90 CONTINUE
C
C
          GO TO 2
C
       91 DO 92 K=1,NSAM
C
          DO 92 I=1,N
          PMUKS2(I,K)=MUKS(I,K)
C
          DO 92 J=1,N
          PKS2(I,J,K)=PKS(I,J,K)
       92 CONTINUE
C
          CALL PROB
       93 CONTINUE
C                                                                    EVAL0364
C         PRINT OUT RESULTS                                          EVAL0365
          WRITE (6,75)                                               EVAL0366
C                                                                    EVAL0367
```

```
      DO 51 K=1,NSAM                                                    EVAL0368
      WRITE (6,79) K                                                    EVAL0369
      WRITE (6,80) (MUKS(I,K),I=1,N)                                    EVAL0370
      WRITE (6,76)                                                      EVAL0371
C                                                                       EVAL0372
      DO 50 I=1,N                                                       EVAL0373
   50 WRITE (6,81) (PKS(I,J,K),J=1,N)                                   EVAL0374
C                                                                       EVAL0375
   51 WRITE (6,82)                                                      EVAL0376
C                                                                       EVAL0377
C     ****************************************************             EVAL0378
C     IF.IPCH.NE.0 MEANS AND COVARIANCES ARE PUNCHED                    EVAL0379
      READ (5,77) IPCH                                                  EVAL0380
      IF (IPCH.EQ.0) GO TO 54                                           EVAL0381
C                                                                       EVAL0382
C                                                                       EVAL0383
      DO 52 K=1,NSAM                                                    EVAL0384
   52 WRITE (7,78) (MUKS(I,K),I=1,N)                                    EVAL0385
C                                                                       EVAL0386
C                                                                       EVAL0387
      DO 53 K=1,NSAM                                                    EVAL0388
C                                                                       EVAL0389
      DO 53 I=1,N                                                       EVAL0390
   53 WRITE (7,78) (PKS(I,J,K),J=1,N)                                   EVAL0391
C                                                                       EVAL0392
   54 CONTINUE                                                          EVAL0393
C                                                                       EVAL0394
C     PLOT OUTPUT
      CALL OPLOT (MUKS,PKS,GKS,NSAM,N,NA)
      STOP                                                              EVAL0396
C                                                                       EVAL0397
   55 FORMAT (8I10)                                                     EVAL0398
   56 FORMAT (8F10.0)                                                   EVAL0399
   57 FORMAT (30X,'A PARTIAL LISTING OF INPUT DATA FOLLOWS',/)          EVAL0400
   58 FORMAT (2X'N = ',I1,2X'L = ',I1,2X,'NA = ',I1,2X'INOISE = ',I1,   EVAL0401
     1 2X,'NSAM = ',I3,//)                                              EVAL0402
   59 FORMAT (4F20.0)                                                   EVAL0403
   60 FORMAT (//,2X,'THE Q MATRIX USED IN THE GAIN CALCULATION IS',/)   EVAL0404
   61 FORMAT (//,2X,'THE R MATRIX USED IN THE GAIN CALCULATION IS',/)   EVAL0405
   62 FORMAT (2X'GAIN MATRICES FOR FIRST AND LAST SAMPLES ARE',/)       EVAL0406
   63 FORMAT (9(2X,1PD12.5),/)                                          EVAL0407
   64 FORMAT (///,1X,'THE THEORETICAL COV OF EST ERROR MATRIX CALC BY   EVAL0408
     1 "GAIN" FOR FIRST AND LAST SAMPLE ARE',/)                         EVAL0409
   65 FORMAT (///,1X,'STATE TRAJ. FOR FIRST AND LAST SAMPLE ARE',/)     EVAL0410
   66 FORMAT (///,1X,'INITIAL STATE ESTIMATE IS',/)                     EVAL0411
   67 FORMAT (///,1X,'THE PHI MATRIX IS',/)                             EVAL0412
   68 FORMAT (///,1X,'THE H MATRIX IS',/)                               EVAL0413
   69 FORMAT (///,1X,'THE C MATRIX IS',/)                               EVAL0414
```

```
   70 FORMAT (///,1X,'THE MATRIX MO(K) IS',/)                         EVAL0415
   71 FORMAT (///,1X,'THE MATRIX M1(K) IS',/)                         EVAL0416
   72 FORMAT (///,1X,'THE MATRIX M2(K) IS',/)                         EVAL0417
   73 FORMAT (///,1X,'THE MATRIX R IS',/)                             EVAL0418
   74 FORMAT ('1')                                                    EVAL0419
   75 FORMAT (2X,'MEANS AND COVARIANCES OF EST. ERROR',//)            EVAL0420
   76 FORMAT (//,1X,'THE COVARIANCE OF EST. ERROR MATRIX IS',/)       EVAL0421
   77 FORMAT (I1)                                                     EVAL0422
   78 FORMAT (4(1PD20.12))                                            EVAL0423
   79 FORMAT (5X,'K = ',I3,/)                                         EVAL0424
   80 FORMAT (1X,'THE ESTIMATION ERROR MEAN IS',/,9(1X,D12.5))        EVAL0425
   81 FORMAT (9(1X,D12.5))                                            EVAL0426
   82 FORMAT (2X,//)                                                  EVAL0427
   83 FORMAT(///,1X,'THE P(0/-1) MATRIX IS',/)
      END                                                             EVAL0428
      SUBROUTINE ADD (A,B,N,M,C)                                      EVAL0429
C     THIS SUBROUTINE ADDS THE NXM MATRICES A AND B, STORING THE      EVAL0430
C     RESULT IN C                                                     EVAL0431
      REAL*8 A,B,C                                                    EVAL0432
      DIMENSION A(9,9),B(9,9),C(9,9)
C                                                                     EVAL0434
      DO 1 I=1,N                                                      EVAL0435
C                                                                     EVAL0436
      DO 1 J=1,M                                                      EVAL0437
    1 C(I,J) = A(I,J)+B(I,J)                                          EVAL0438
C                                                                     EVAL0439
      RETURN                                                          EVAL0440
      END                                                             EVAL0441
      SUBROUTINE A1                                                   EVAL0442
      IMPLICIT REAL*8 (A-H,O-Z)                                       EVAL0443
      REAL*8 MOK,M1K,M2K,MUK,MUKS                                     EVAL0444
      COMMON GKS(9,9,60),PKS(9,9,60),XKS(9,60),MUKS(9,60),SOK(9,9),
     1S1K(9,9),S2K(9,9),S3K(9,9),PHI(9,9),C(9,9),H(9,9),EID(9,9),
     2A1K(9,9),A2K(9,9),GK(9,9),MOK(9,9),M1K(9,9),M2K(9,9),SOKM1(9,9),
     3S1KM1(9,9),RK(9,9),SOKM2(9,9),RKM1(9,9),RKM2(9,9),S3KM1(9,9),
     4TMP1(9,9),TMP2(9,9),PK(9,9),XK(9),VTMP1(9),XHIT(9),VTMP2(9),
     5CON(9),VTMP3(9),MUK(9),XKM1(9),XKM2(9),DK(9),N,L,NA,K,ND,NSAM,
     6IP,IP1,IH,IQ,PHK(9,9,60),HK(9,9,60),QK(9,9,60),PMUKS1(9,60),
     7PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)
C     THIS SUBROUTINE COMPUTES THE CURRENT VALUE OF THE               EVAL0451
C     NXN MATRIX A1(K) FROM EQ. (4.33)                                EVAL0452
C                                                                     EVAL0453
C                                                                     EVAL0454
      CALL TRANS (S1K,N,NA,TMP1)                                      EVAL0455
      CALL PROD (RKM1,TMP1,NA,NA,N,TMP2)                              EVAL0456
      CALL PROD (SOKM1,TMP2,N,NA,N,TMP1)                              EVAL0457
      CALL PROD (S3K,TMP1,N,N,N,A1K)                                  EVAL0458
      RETURN                                                          EVAL0459
```

```
      END                                                               EVAL0460
      SUBROUTINE A2                                                     EVAL0461
      IMPLICIT REAL*8 (A-H,O-Z)                                         EVAL0462
      REAL*8 MOK,M1K,M2K,MUK,MUKS                                       EVAL0463
      COMMON GKS(9,9,60),PKS(9,9,60),XKS(9,60),MUKS(9,60),SOK(9,9),
     1S1K(9,9),S2K(9,9),S3K(9,9),PHI(9,9),C(9,9),H(9,9),EID(9,9),
     2A1K(9,9),A2K(9,9),GK(9,9),MOK(9,9),M1K(9,9),M2K(9,9),SOKM1(9,9),
     3S1KM1(9,9),RK(9,9),SOKM2(9,9),RKM1(9,9),RKM2(9,9),S3KM1(9,9),
     4TMP1(9,9),TMP2(9,9),PK(9,9),XK(9),VTMP1(9),XHIT(9),VTMP2(9),
     5CON(9),VTMP3(9),MUK(9),XKM1(9),XKM2(9),DK(9),N,L,NA,K,ND,NSAM,
     6IP,IP1,IH,IQ,PHK(9,9,60),HK(9,9,60),QK(9,9,60),PMUKS1(9,60),
     7PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)
C       THIS SUBROUTINE COMPUTES THE CURRENT VALUE OF THE               EVAL0470
C      NXN MATRIX A2(K)                                                 EVAL0471
      CALL TRANS (S2K,N,NA,TMP1)                                        EVAL0472
      CALL PROD (RKM2,TMP1,NA,NA,N,TMP2)                                EVAL0473
      CALL PROD (S3KM1,SOKM2,N,N,NA,TMP1)                               EVAL0474
      CALL ADD (TMP1,S1KM1,N,NA,A2K)                                    EVAL0475
      CALL PROD (A2K,TMP2,N,NA,N,TMP1)                                  EVAL0476
      CALL PROD (S3K,TMP1,N,N,N,A2K)                                    EVAL0477
      RETURN                                                            EVAL0478
      END                                                               EVAL0479
      SUBROUTINE COVE                                                   EVAL0480
      IMPLICIT REAL*8 (A-H,O-Z)                                         EVAL0481
      REAL*8 MOK,M1K,M2K,MUK,MUKS                                       EVAL0482
      COMMON GKS(9,9,60),PKS(9,9,60),XKS(9,60),MUKS(9,60),SOK(9,9),
     1S1K(9,9),S2K(9,9),S3K(9,9),PHI(9,9),C(9,9),H(9,9),EID(9,9),
     2A1K(9,9),A2K(9,9),GK(9,9),MOK(9,9),M1K(9,9),M2K(9,9),SOKM1(9,9),
     3S1KM1(9,9),RK(9,9),SOKM2(9,9),RKM1(9,9),RKM2(9,9),S3KM1(9,9),
     4TMP1(9,9),TMP2(9,9),PK(9,9),XK(9),VTMP1(9),XHIT(9),VTMP2(9),
     5CON(9),VTMP3(9),MUK(9),XKM1(9),XKM2(9),DK(9),N,L,NA,K,ND,NSAM,
     6IP,IP1,IH,IQ,PHK(9,9,60),HK(9,9,60),QK(9,9,60),PMUKS1(9,60),
     7PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)
C      THIS SUBROUTINE CALCULATES THE COVARIANCE OF ESTIMATION          EVAL0489
C      ERROR USING EQ. (4.33)                                           EVAL0490
C                                                                       EVAL0491
      CALL ADD (A1K,A2K,N,N,A1K)                                        EVAL0492
      CALL TRANS (A1K,N,N,A2K)                                          EVAL0493
      CALL ADD (A1K,A2K,N,N,A1K)                                        EVAL0494
C     A1K NOW CONTAINS A1K+A2K + TRANSPOSE(A1K+A2K)                     EVAL0495
      CALL TRANS (S2K,N,NA,TMP1)                                        EVAL0496
      CALL PROD (RKM2,TMP1,NA,NA,N,TMP2)                                EVAL0497
      CALL PROD (S2K,TMP2,N,NA,N,A2K)                                   EVAL0498
      CALL ADD (A2K,A1K,N,N,A2K)                                        EVAL0499
      CALL TRANS (S1K,N,NA,TMP1)                                        EVAL0500
      CALL PROD (RKM1,TMP1,NA,NA,N,TMP2)                                EVAL0501
      CALL PROD (S1K,TMP2,N,NA,N,TMP1)                                  EVAL0502
      CALL ADD (A2K,TMP1,N,N,A2K)                                       EVAL0503
```

```
      CALL TRANS (SOK,N,NA,TMP1)                                        EVAL0504
      CALL PROD (RK,TMP1,NA,NA,N,TMP2)                                  EVAL0505
      CALL PROD (SOK,TMP2,N,NA,N,TMP1)                                  EVAL0506
      CALL ADD (A2K,TMP1,N,N,A2K)                                       EVAL0507
      CALL TRANS (S3K,N,N,TMP1)                                         EVAL0508
      CALL PROD (PK,TMP1,N,N,N,TMP2)                                    EVAL0509
      CALL PROD (S3K,TMP2,N,N,N,TMP1)                                   EVAL0510
      CALL ADD (A2K,TMP1,N,N,PK)                                        EVAL0511
      RETURN                                                            EVAL0512
      END                                                               EVAL0513
      SUBROUTINE D                                                      EVAL0514
      IMPLICIT REAL*8 (A-H,O-Z)                                         EVAL0515
      REAL*8 MOK,M1K,M2K,MUK,MUKS                                       EVAL0516
      COMMON GKS(9,9,60),PKS(9,9,60),XKS(9,60),MUKS(9,60),SOK(9,9),
     1S1K(9,9),S2K(9,9),S3K(9,9),PHI(9,9),C(9,9),H(9,9),EID(9,9),
     2A1K(9,9),A2K(9,9),GK(9,9),MOK(9,9),M1K(9,9),M2K(9,9),SOKM1(9,9),
     3S1KM1(9,9),RK(9,9),SOKM2(9,9),RKM1(9,9),RKM2(9,9),S3KM1(9,9),
     4TMP1(9,9),TMP2(9,9),PK(9,9),XK(9),VTMP1(9),XHIT(9),VTMP2(9),
     5CON(9),VTMP3(9),MUK(9),XKM1(9),XKM2(9),DK(9),N,L,NA,K,ND,NSAM,
     6IP,IP1,IH,IQ,PHK(9,9,60),HK(9,9,60),QK(9,9,60),PMUKS1(9,60),
     7PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)
C     THIS SUBROUTINE COMPUTES THE CURRENT VALUE OF THE N-VECTOR        EVAL0523
C     D(K) FROM EQ.(4.10)                                               EVAL0524
C                                                                       EVAL0525
      IF(IH.NE.0) GO TO 1
      DO 1 I=1,NA
C
      DO 1 J=1,N
      H(I,J)=HK(I,J,K)
    1 CONTINUE
C                                                                       EVAL0526
      CALL VPROD (H,XKM2,NA,N,VTMP1)                                    EVAL0527
      CALL VPROD (S2K,VTMP1,N,NA,VTMP1)                                 EVAL0528
      CALL VPROD (H,XKM1,NA,N,DK)                                       EVAL0529
      CALL VPROD (S1K,DK,N,NA,DK)                                       EVAL0530
      CALL VADD (DK,VTMP1,N,DK)                                         EVAL0531
      CALL PROD (SOK,H,N,NA,N,TMP1)                                     EVAL0532
      CALL SUB (TMP1,EID,N,N,TMP2)                                      EVAL0533
      CALL VPROD (TMP2,XK,N,N,VTMP1)                                    EVAL0534
      CALL VADD (VTMP1,DK,N,DK)                                         EVAL0535
      IF(IP.NE.0) GO TO 2
      CALL UK
      CALL VADD (DK,VTMP2,N,DK)
    2 CONTINUE
      RETURN                                                            EVAL0536
      END                                                               EVAL0537
      SUBROUTINE GAIN                                                   EVAL0538
      IMPLICIT REAL*8 (A-H,O-Z)                                         EVAL0539
```

```
      REAL*8 MOK,M1K,M2K,MUK,MUKS                                       EVAL0540
      COMMON GKS(9,9,60),PKS(9,9,60),XKS(9,60),MUKS(9,60),SOK(9,9),
     1S1K(9,9),S2K(9,9),S3K(9,9),PHI(9,9),C(9,9),H(9,9),EID(9,9),
     2A1K(9,9),A2K(9,9),GK(9,9),MOK(9,9),M1K(9,9),M2K(9,9),SOKM1(9,9),
     3S1KM1(9,9),RK(9,9),SOKM2(9,9),RKM1(9,9),RKM2(9,9),S3KM1(9,9),
     4TMP1(9,9),TMP2(9,9),PK(9,9),XK(9),VTMP1(9),XHIT(9),VTMP2(9),
     5CON(9),VTMP3(9),MUK(9),XKM1(9),XKM2(9),DK(9),N,L,NA,K,ND,NSAM,
     6IP,IP1,IH,IQ,PHK(9,9,60),HK(9,9,60),QK(9,9,60),PMUKS1(9,60),
     7PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)
C                                                                       EVAL0547
C     G(K) = P(K/K-1)*HT*(H*P(K/K-1)*HT + R)                            EVAL0548
C                                                                       EVAL0549
      DO 6 K=1,NSAM                                                     EVAL0550
C
      IF(IH.NE.0) GO TO 7
      DO 7 I=1,NA
C
      DO 7 J=1,N
      H(I,J)=HK(I,J,K)
    7 CONTINUE
C
      IF(IP.NE.0) GO TO 8
C
      DO 8 I=1,N
      DO 8 J=1,N
      PHI(I,J)=PHK(I,J,K)
    8 CONTINUE
      IF(IQ.NE.0)GO TO 9
C
      DO 9 I=1,N
      DO 9 J=1,N
C
      RKM2(I,J)=QK(I,J,K)
    9 CONTINUE
      CALL TRANS (H,NA,N,TMP2)                                          EVAL0551
      CALL PROD (PK,TMP2,N,N,NA,SOK)                                    EVAL0552
      CALL PROD (H,SOK,NA,N,NA,TMP1)                                    EVAL0553
      CALL ADD (TMP1,RKM1,NA,NA,TMP1)                                   EVAL0554
      IF (NA.EQ.1) GO TO 4                                              EVAL0555
      MD = ND                                                           EVAL0556
      CALL GAUSS3 (NA,EPS,TMP1,TMP2,KER,MD)                             EVAL0557
      CALL PROD (SOK,TMP2,N,NA,NA,GK)                                   EVAL0558
C                                                                       EVAL0559
C     NOTE HERE  PKK(I,J) = P(K/K)  WHERE                               EVAL0560
C     P(K/K) = (I-G(K)*H)*P(K/K-1)                                      EVAL0561
    1 CALL PROD (GK,H,N,NA,N,SOK)                                       EVAL0562
      CALL SUB (FID,SOK,N,N,TMP2)                                       EVAL0563
      CALL PROD (TMP2,PK,N,N,N,S1K)                                     EVAL0564
```

```
C                                                                       EVAL0565
C        NOTE HERE   PKKM1(I,J) = P(K/K-1)   WHERE                      EVAL0566
C        P(K/K-1)= PHI*P(K-1/K-1)*PHIT + Q                              EVAL0567
         CALL TRANS (PHI,N,N,TMP2)                                      EVAL0568
         CALL PROD (S1K,TMP2,N,N,N,SOK)                                 EVAL0569
         CALL PROD (PHI,SOK,N,N,N,TMP1)                                 EVAL0570
         CALL ADD (TMP1,RKM2,N,N,PK)                                    EVAL0571
C                                                                       EVAL0572
         DO 3 I=1,N                                                     EVAL0573
C                                                                       EVAL0574
         DO 2 J=1,NA                                                    EVAL0575
       2 GKS(I,J,K) = GK(I,J)                                           EVAL0576
C                                                                       EVAL0577
C                                                                       EVAL0578
         DO 3 JK=1,N                                                    EVAL0579
       3 PKS(I,JK,K) = S1K(I,JK)                                        EVAL0580
C                                                                       EVAL0581
         GO TO 6                                                        EVAL0582
C                                                                       EVAL0583
       4 DO 5 I=1,N                                                     EVAL0584
       5 GK(I,1) = SOK(I,1)/TMP1(1,1)                                   EVAL0585
C                                                                       EVAL0586
         GO TO 1                                                        EVAL0587
       6 CONTINUE                                                       EVAL0588
C                                                                       EVAL0589
         RETURN                                                         EVAL0590
         END                                                            EVAL0591
         SUBROUTINE LOADG                                               EVAL0592
         IMPLICIT REAL*8 (A-H,O-Z)                                      EVAL0593
         REAL*8 MOK,M1K,M2K,MUK,MUKS                                    EVAL0594
         COMMON GKS(9,9,60),PKS(9,9,60),XKS(9,60),MUKS(9,60),SOK(9,9),
        1S1K(9,9),S2K(9,9),S3K(9,9),PHI(9,9),C(9,9),H(9,9),EID(9,9),
        2A1K(9,9),A2K(9,9),GK(9,9),MOK(9,9),M1K(9,9),M2K(9,9),SOKM1(9,9),
        3S1KM1(9,9),RK(9,9),SOKM2(9,9),RKM1(9,9),RKM2(9,9),S3KM1(9,9),
        4TMP1(9,9),TMP2(9,9),PK(9,9),XK(9),VTMP1(9),XHIT(9),VTMP2(9),
        5CON(9),VTMP3(9),MUK(9),XKM1(9),XKM2(9),DK(9),N,L,NA,K,ND,NSAM,
        6IP,IP1,IH,IQ,PHK(9,9,60),HK(9,9,60),QK(9,9,60),PMUKS1(9,60),
        7PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)
C                                                                       EVAL0601
         DO 1 I=1,N                                                     EVAL0602
C                                                                       EVAL0603
         DO 1 J=1,L                                                     EVAL0604
       1 GK(I,J) = GKS(I,J,K)                                           EVAL0605
C                                                                       EVAL0606
         RETURN                                                         EVAL0607
         END                                                            EVAL0608
         SUBROUTINE MEANE                                               EVAL0609
C        THIS SUBROUTINE CALCULATES THE MEAN OF ESTIMATION ERROR        EVAL0610
```

```
C     FROM EQ. (4.34)                                                   EVAL0611
      IMPLICIT REAL*8 (A-H,O-Z)                                         EVAL0612
      REAL*8 MOK,M1K,M2K,MUK,MUKS                                       EVAL0613
      COMMON GKS(9,9,60),PKS(9,9,60),XKS(9,60),MUKS(9,60),SOK(9,9),
     1S1K(9,9),S2K(9,9),S3K(9,9),PHI(9,9),C(9,9),H(9,9),EID(9,9),
     2A1K(9,9),A2K(9,9),GK(9,9),MOK(9,9),M1K(9,9),M2K(9,9),SOKM1(9,9),
     3S1KM1(9,9),RK(9,9),SOKM2(9,9),RKM1(9,9),RKM2(9,9),S3KM1(9,9),
     4TMP1(9,9),TMP2(9,9),PK(9,9),XK(9),VTMP1(9),XHIT(9),VTMP2(9),
     5CON(9),VTMP3(9),MUK(9),XKM1(9),XKM2(9),DK(9),N,L,NA,K,ND,NSAM,
     6IP,IP1,IH,IQ,PHK(9,9,60),HK(9,9,60),QK(9,9,60),PMUKS1(9,60),
     7PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)
      CALL VADD (MUK,XKM1,N,MUK)                                        EVAL0620
      CALL VPROD (S3K,MUK,N,N,MUK)                                      EVAL0621
      CALL VADD (MUK,DK,N,MUK)                                          EVAL0622
      RETURN                                                            EVAL0623
      END                                                               EVAL0624
      SUBROUTINE PROD (A,B,N,M,L,C)                                     EVAL0625
C     THIS SUBROUTINE COMPUTES THE MATRIX PRODUCT AB AND STORES THE     EVAL0626
C     RESULT IN C                                                       EVAL0627
C     A = NXM,    B = MXL,    C = NXL                                   EVAL0628
      REAL*8 A,B,C,T                                                    EVAL0629
      DIMENSION A(9,9),B(9,9),C(9,9),T(9,9)
C                                                                       EVAL0631
      DO 1 I=1,N                                                        EVAL0632
C                                                                       EVAL0633
      DO 1 J=1,L                                                        EVAL0634
    1 T(I,J) = 0.                                                       EVAL0635
C                                                                       EVAL0636
C                                                                       EVAL0637
      DO 2 I=1,N                                                        EVAL0638
C                                                                       EVAL0639
      DO 2 J=1,L                                                        EVAL0640
C                                                                       EVAL0641
      DO 2 K=1,M                                                        EVAL0642
    2 T(I,J) = T(I,J)+A(I,K)*B(K,J)                                     EVAL0643
C                                                                       EVAL0644
C                                                                       EVAL0645
      DO 3 I=1,N                                                        EVAL0646
C                                                                       EVAL0647
      DO 3 J=1,L                                                        EVAL0648
    3 C(I,J) = T(I,J)                                                   EVAL0649
C                                                                       EVAL0650
      RETURN                                                            EVAL0651
      END                                                               EVAL0652
      SUBROUTINE RNOISE                                                 EVAL0653
      IMPLICIT REAL*8 (A-H,O-Z)                                         EVAL0654
      REAL*8 MOK,M1K,M2K,MUK,MUKS                                       EVAL0655
      COMMON GKS(9,9,60),PKS(9,9,60),XKS(9,60),MUKS(9,60),SOK(9,9),
```

```
     1S1K(9,9),S2K(9,9),S3K(9,9),PHI(9,9),C(9,9),H(9,9),EID(9,9),
     2A1K(9,9),A2K(9,9),GK(9,9),MOK(9,9),M1K(9,9),M2K(9,9),SOKM1(9,9),
     3S1KM1(9,9),RK(9,9),SOKM2(9,9),RKM1(9,9),RKM2(9,9),S3KM1(9,9),
     4TMP1(9,9),TMP2(9,9),PK(9,9),XK(9),VTMP1(9),XHIT(9),VTMP2(9),
     5CON(9),VTMP3(9),MUK(9),XKM1(9),XKM2(9),DK(9),N,L,NA,K,ND,NSAM,
     6IP,IP1,IH,IQ,PHK(9,9,60),HK(9,9,60),QK(9,9,60),PMUKS1(9,60),
     7PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)
      RANGE = DSQRT(XKS(1,K)**2+XKS(3,K)**2)                            EVAL0662
      BEAR = DATAN2(XKS(3,K),XKS(1,K))                                  EVAL0663
      VARR = CON(1)**2                                                  EVAL0664
      VARB = CON(2)**2                                                  EVAL0665
      ALPHA = (RANGE*CON(2))**2                                         EVAL0666
      SIBR = DSIN(BEAR)                                                 EVAL0667
      COBR = DCOS(BEAR)                                                 EVAL0668
      RK(1,1) = ALPHA*SIBR**2+VARR*COBR**2                              EVAL0669
      RK(1,2) = (VARR-ALPHA)*COBR*SIBR                                  EVAL0670
      RK(2,1) = RK(1,2)                                                 EVAL0671
      RK(2,2) = ALPHA*COBR**2+VARR*SIBR**2                              EVAL0672
      RETURN                                                            EVAL0673
      END                                                               EVAL0674
      SUBROUTINE STORE                                                  EVAL0675
      IMPLICIT REAL*8 (A-H,O-Z)                                         EVAL0676
      REAL*8 MOK,M1K,M2K,MUK,MUKS                                       EVAL0677
      COMMON GKS(9,9,60),PKS(9,9,60),XKS(9,60),MUKS(9,60),SOK(9,9),
     1S1K(9,9),S2K(9,9),S3K(9,9),PHI(9,9),C(9,9),H(9,9),EID(9,9),
     2A1K(9,9),A2K(9,9),GK(9,9),MOK(9,9),M1K(9,9),M2K(9,9),SOKM1(9,9),
     3S1KM1(9,9),RK(9,9),SOKM2(9,9),RKM1(9,9),RKM2(9,9),S3KM1(9,9),
     4TMP1(9,9),TMP2(9,9),PK(9,9),XK(9),VTMP1(9),XHIT(9),VTMP2(9),
     5CON(9),VTMP3(9),MUK(9),XKM1(9),XKM2(9),DK(9),N,L,NA,K,ND,NSAM,
     6IP,IP1,IH,IQ,PHK(9,9,60),HK(9,9,60),QK(9,9,60),PMUKS1(9,60),
     7PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)
C     THIS SUBROUTINE STORES P(K),MU(K) AFTER THEY HAVE BEEN            EVAL0684
C     COMPUTED                                                          EVAL0685
C                                                                       EVAL0686
C                                                                       EVAL0687
C                                                                       EVAL0688
      DO 1 I=1,N                                                        EVAL0689
      MUKS(I,K) = MJK(I)                                                EVAL0690
C                                                                       EVAL0691
      DO 1 J=1,N                                                        EVAL0692
    1 PKS(I,J,K) = PK(I,J)                                              EVAL0693
C                                                                       EVAL0694
      RETURN                                                            EVAL0695
      END                                                               EVAL0696
      SUBROUTINE SUB (A,B,N,M,C)                                        EVAL0697
C     THIS SUBROUTINE SUBTRACTS THE NXM MATRIX B FROM THE NXM MATRIX    EVAL0698
C     A AND STORES THE RESULT IN C                                      EVAL0699
      REAL*8 A,B,C                                                      EVAL0700
```

```
      DIMENSION A(9,9),B(9,9),C(9,9)
C                                                                       EVAL0702
      DO 1 I=1,N                                                        EVAL0703
C                                                                       EVAL0704
      DO 1 J=1,M                                                        EVAL0705
    1 C(I,J) = A(I,J)-B(I,J)                                            EVAL0706
C                                                                       EVAL0707
      RETURN                                                            EVAL0708
      END                                                               EVAL0709
      SUBROUTINE S3                                                     EVAL0710
      IMPLICIT REAL*8 (A-H,O-Z)                                         EVAL0711
      REAL*8 MOK,M1K,M2K,MUK,MUKS                                       EVAL0712
      COMMON GKS(9,9,60),PKS(9,9,60),XKS(9,60),MUKS(9,60),SOK(9,9),
     1S1K(9,9),S2K(9,9),S3K(9,9),PHI(9,9),C(9,9),H(9,9),EID(9,9),
     2A1K(9,9),A2K(9,9),GK(9,9),MOK(9,9),M1K(9,9),M2K(9,9),SOKM1(9,9),
     3S1KM1(9,9),RK(9,9),SOKM2(9,9),RKM1(9,9),RKM2(9,9),S3KM1(9,9),
     4TMP1(9,9),TMP2(9,9),PK(9,9),XK(9),VTMP1(9),XHIT(9),VTMP2(9),
     5CON(9),VTMP3(9),MUK(9),XKM1(9),XKM2(9),DK(9),N,L,NA,K,ND,NSAM,
     6IP,IP1,IH,IQ,PHK(9,9,60),HK(9,9,60),QK(9,9,60),PMUKS1(9,60),
     7PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)
C
      IF(IP.NE.0) GO TO 1
      DO 1 I=1,N
C
      DO 1 J=1,N
      PHI(I,J)=PHK(I,J,K-1)
    1 CONTINUE
C
      IF(IH.NE.0) GO TO 2
C
      DO 2 I=1,NA
C
      DO 2 J=1,N
      C(I,J)=HK(I,J,K)
    2 CONTINUE
      CALL PROD (GK,C,N,L,N,TMP1)                                       EVAL0719
      CALL SUB (EID,TMP1,N,N,TMP1)                                      EVAL0720
      CALL PROD (TMP1,PHI,N,N,N,S3K)                                    EVAL0721
      RETURN                                                            EVAL0722
      END                                                               EVAL0723
      SUBROUTINE TRACK                                                  EVAL0724
C     IF TRACK IS TO BE GENERATED ON-LINE IT IS DONE IN THIS SUBROUTINE EVAL0725
C                                                                       EVAL0726
       IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 MOK,M1K,M2K,MUK,MUKS
      COMMON GKS(9,9,60),PKS(9,9,60),XKS(9,60),MUKS(9,60),SOK(9,9),
     1S1K(9,9),S2K(9,9),S3K(9,9),PHI(9,9),C(9,9),H(9,9),EID(9,9),
     2A1K(9,9),A2K(9,9),GK(9,9),MOK(9,9),M1K(9,9),M2K(9,9),SOKM1(9,9),
```

```
     3S1KM1(9,9),RK(9,9),SOKM2(9,9),RKM1(9,9),RKM2(9,9),S3KM1(9,9),
     4TMP1(9,9),TMP2(9,9),PK(9,9),XK(9),VTMP1(9),XHIT(9),VTMP2(9),
     5CON(9),VTMP3(9),MUK(9),XKM1(9),XKM2(9),DK(9),N,L,NA,K,ND,NSAM,
     6IP,IP1,IH,IQ,PHK(9,9,60),HK(9,9,60),QK(9,9,60),PMUKS1(9,60),
     7PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)
C
C
      XKS(1,1)=2992.59D0
      XKS(2,1)=-5295.85D0
      XKS(3,1)=4.504D0
      XKS(4,1)=5.06145D0
      XKS(5,1)=500.D0
C
      DO 3 K=2,NSAM
C
      DO 1 I=1,N
      TMP1(I,1)=XKS(I,K-1)
    1 CONTINUE
C
      CALL PROD(PHI,TMP1,N,N,1,TMP1)
      DO 2 I=1,N
    2 XKS(I,K)=TMP1(I,1)
C
    3 CONTINUE
       RETURN
       END
      SUBROUTINE TRANS (A,N,M,C)                                        EVAL0746
C     THIS SUBROUTINE FORMS THE MATRIX TRANSPOSE OF A STORING THE       EVAL0747
C     RESULT IN C                                                       EVAL0748
C     A = NXM,    C = MXN                                               EVAL0749
      REAL*8 A,C                                                        EVAL0750
      DIMENSION A(9,9),C(9,9)
C                                                                       EVAL0752
      DO 1 I=1,N                                                        EVAL0753
C                                                                       EVAL0754
      DO 1 J=1,M                                                        EVAL0755
    1 C(J,I) = A(I,J)                                                   EVAL0756
C                                                                       EVAL0757
      RETURN                                                            EVAL0758
      END                                                               EVAL0759
      SUBROUTINE VADD (X,Y,N,Z)                                         EVAL0760
C     THIS SUBROUTINE COMPUTES THE SUM OF THE N-VECTORS X AND           EVAL0761
C     Y AND STORES THE RESULT IN THE N-VECTOR Z                         EVAL0762
C                                                                       EVAL0763
      REAL*8 X(9),Y(9),Z(9)
C                                                                       EVAL0765
      DO 1 I=1,N                                                        EVAL0766
    1 Z(I) = X(I)+Y(I)                                                  EVAL0767
```

```
C
      RETURN
      END
      SUBROUTINE VPROD (A,X,M,N,Y)
C     THIS SUBROUTINE COMPUTES THE PRODUCT OF THE MXN MATRIX
C     A AND THE N-VECTOR X AND STORES THE RESULT IN THE
C     M-VECTOR Y
C
C
      REAL*8 A(9,9),X(9),Y(9),T(9)
C
      DO 1 I=1,M
      T(I) = 0.D0
C
      DO 1 J=1,N
    1 T(I) = T(I)+A(I,J)*X(J)
C
C
      DO 2 I=1,M
    2 Y(I) = T(I)
C
      RETURN
      END
      SUBROUTINE VSUB (X,Y,N,Z)
C     THIS SUBROUTINE COMPUTES THE DIFFERENCE X-Y OF THE TWO
C     N-VECTORS X & Y AND STORES THE RESULT IN THE N-VECTOR Z
C
      REAL*8 X(9),Y(9),Z(9)
C
      DO 1 I=1,N
    1 Z(I) = X(I)-Y(I)
C
      RETURN
      END
      SUBROUTINE OPLOT (MUKS,PKS,GKS,NSAM,N,NA)
C
C     THIS SUBROUTINEPLOTS THE OUTPUT
C
      REAL*8 MUKS,PKS,GKS
C
      DIMENSION MUKS(9,60),PKS(9,9,60),GKS(9,9,60),XP(60),YP(60)
C
C
C
      DO 88 K=1,NSAM
   88 XP(K)=K
C
      DO 90 I=1,N
```

```
C
      DO 89 K=1,NSAM
C
   89 YP(K)=MUKS(I,K)
      WRITE (6,85)
      CALL PLOTP (XP,YP,NSAM,0)
C
   90 WRITE (6,86) I,I
C
C
      DO 92 I=1,N
C
      DO 91 K=1,NSAM
C
   91 YP(K)=PKS(I,I,K)
C
      WRITE(6,85)
      CALL PLOTP (XP,YP,NSAM,0)
C
   92 WRITE (6,87) I
C
      DO 94 I=1,N
      DO 94 J=I,NA
      DO 93 K=1,NSAM
   93 YP(K)=GKS(I,J,K)
C
      WRITE (6,85)
      CALL PLOTP (XP,YP,NSAM,0)
   94 WRITE (6,95) I,J
   95 FORMAT (12X,'G(',I1,',',I1,')  VS.  K')
   85 FORMAT ('1')
   86 FORMAT (12X,'XHAT(',I1,')-X(',I1,')  VS      K')
   87 FORMAT (12X,'ERROR VARIANCE(',I1,')  VS     K')
      RETURN
      END
      SUBROUTINE HKK
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 MOK,M1K,M2K,MUK,MUKS
      COMMON GKS(9,9,60),PKS(9,9,60),XKS(9,60),MUKS(9,60),SOK(9,9),
     1S1K(9,9),S2K(9,9),S3K(9,9),PHI(9,9),C(9,9),H(9,9),EID(9,9),
     2A1K(9,9),A2K(9,9),GK(9,9),MOK(9,9),M1K(9,9),M2K(9,9),SOKM1(9,9),
     3S1KM1(9,9),RK(9,9),SOKM2(9,9),RKM1(9,9),RKM2(9,9),S3KM1(9,9),
     4TMP1(9,9),TMP2(9,9),PK(9,9),XK(9),VTMP1(9),XHIT(9),VTMP2(9),
     5CON(9),VTMP3(9),MUK(9),XKM1(9),XKM2(9),DK(9),N,L,NA,K,ND,NSAM,
     6IP,IP1,IH,IQ,PHK(9,9,60),HK(9,9,60),QK(9,9,60),PMUKS1(9,60),
     7PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)
C
```

```
C     THIS SUBROUTINE COMPUTES LINEARIZED MEASUREMENT MATRIX H(K)=HK(I,J,K)
C
C
      A=-1.D0
      VP=1640.D0
C
      DO 1 K=1,NSAM
C
      R=(XKS(1,K)**2)+(XKS(2,K)**2)
      R=DSQRT(R)
C
      F=VP+(XKS(3,K)*XKS(2,K))/R
      F=(XKS(5,K)*VP)/F
C
      HK(1,1,K)=-(A*XKS(2,K))/(R**2)
      HK(1,2,K)=(A*XKS(1,K))/(R**2)
      HK(1,3,K)=0.D0
      HK(1,4,K)=1.D0
      HK(1,5,K)=0.D0
C
      HK(2,1,K)=(F**2)*XKS(3,K)*XKS(1,K)*XKS(2,K)
      HK(2,1,K)=HK(2,1,K)/(XKS(5,K)*VP*(R**3))
      HK(2,2,K)=-((F**2)*XKS(3,K)*(XKS(1,K)**2))
      HK(2,2,K)=HK(2,2,K)/((R**3)*VP*XKS(5,K))
      HK(2,3,K)=-((F**2)*XKS(2,K))
      HK(2,3,K)=HK(2,3,K)/(XKS(5,K)*VP*R)
      HK(2,4,K)=0.D0
      HK(2,5,K)=F/XKS(5,K)
C
    1 CONTINUE
      RETURN
      END
      SUBROUTINE AK
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 MOK,M1K,M2K,MUK,MUKS
      COMMON GKS(9,9,60),PKS(9,9,60),XKS(9,60),MUKS(9,60),SOK(9,9),
     1S1K(9,9),S2K(9,9),S3K(9,9),PHI(9,9),C(9,9),H(9,9),EID(9,9),
     2A1K(9,9),A2K(9,9),GK(9,9),MOK(9,9),M1K(9,9),M2K(9,9),SOKM1(9,9),
     3S1KM1(9,9),RK(9,9),SOKM2(9,9),RKM1(9,9),RKM2(9,9),S3KM1(9,9),
     4TMP1(9,9),TMP2(9,9),PK(9,9),XK(9),VTMP1(9),XHIT(9),VTMP2(9),
     5CON(9),VTMP3(9),MUK(9),XKM1(9),XKM2(9),DK(9),N,L,NA,K,ND,NSAM,
     6IP,IP1,IH,IQ,PHK(9,9,60),HK(9,9,60),QK(9,9,60),PMUKS1(9,60),
     7PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)
C
C     THIS SUBROUTINE COMPUTES LINEARIZED STATE MATRIXPH(K)=PHK(I,J,K)
C
C
```

```
C
      DO 1 K=1,NSAM
    1 CONTINUE
C
      RETURN
      END
      SUBROUTINE UK
C
      IMPLICIT REAL*8 (A-H,O-Z)                                         EVAL0727
      REAL*8 MOK,M1K,M2K,MUK,MUKS                                       EVAL0728
      COMMON GKS(9,9,60),PKS(9,9,60),XKS(9,60),MUKS(9,60),SOK(9,9),
     1S1K(9,9),S2K(9,9),S3K(9,9),PHI(9,9),C(9,9),H(9,9),EID(9,9),
     2A1K(9,9),A2K(9,9),GK(9,9),MOK(9,9),M1K(9,9),M2K(9,9),SOKM1(9,9),
     3S1KM1(9,9),RK(9,9),SOKM2(9,9),RKM1(9,9),RKM2(9,9),S3KM1(9,9),
     4TMP1(9,9),TMP2(9,9),PK(9,9),XK(9),VTMP1(9),XHIT(9),VTMP2(9),
     5CON(9),VTMP3(9),MUK(9),XKM1(9),XKM2(9),DK(9),N,L,NA,K,ND,NSAM,
     6IP,IP1,IH,IQ,PHK(9,9,60),HK(9,9,60),QK(9,9,60),PMUKS1(9,60),
     7PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)
C     THIS SUBROUTINE COMPUTES THE FORCING INPUT U(K) WHICH RESULTS FROM
C     LINERIZATION  U(K)=A(X(K),K)-PHK(I,JK)X(K)
C
      DO 1 I=1,N
C
      VTMP1(I)=XKS(I,K-1)
C
      DO 1 J=1,N
    1 TMP1(I,J)=PHK(I,J,K-1)
C
      CALL VPROD (TMP1,VTMP1,N,N,VTMP2)
C
      DO 2 I=1,N
    2 VTMP1(I)=XKS(I,K)
C
      CALL VSUB (VTMP1,VTMP2,N,VTMP2)
C
C     CALCULATE (I-G(K)C(K))U'(K-1)
C
      CALL PROD (GK,C,N,L,N,TMP1)
      CALL SUB (EID,TMP1,N,N,TMP1)
      CALL VPROD (TMP1,VTMP2,N,N,VTMP2)
C
      RETURN
      END
      SUBROUTINE PROB
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 MOK,M1K,M2K,MUK,MUKS
      COMMON GKS(9,9,60),PKS(9,9,60),XKS(9,60),MUKS(9,60),SOK(9,9),
     1S1K(9,9),S2K(9,9),S3K(9,9),PHI(9,9),C(9,9),H(9,9),EID(9,9),
```

```
     2A1K(9,9),A2K(9,9),GK(9,9),MOK(9,9),M1K(9,9),M2K(9,9),SOKM1(9,9),
     3S1KM1(9,9),RK(9,9),SOKM2(9,9),RKM1(9,9),RKM2(9,9),S3KM1(9,9),
     4TMP1(9,9),TMP2(9,9),PK(9,9),XK(9),VTMP1(9),XHIT(9),VTMP2(9),
     5CON(9),VTMP3(9),MUK(9),XKM1(9),XKM2(9),DK(9),N,L,NA,K,ND,NSAM,
     6IP,IP1,IH,IQ,PHK(9,9,60),HK(9,9,60),QK(9,9,60),PMUKS1(9,60),
     7PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)
C       THIS SUBROUTINE CALCULATES MEAN AND COVARIANCE FROM MEANS ANDCOVARIANCES
C      CALCULATED BY USING TWO TRACKS
C
C       CALCULATE MU(K)=MU1(K)*P+MU2(K)*(1-P)
       DO 9 K=1,NSAM
C
C
       DO 1 I=1,N
C
     1 MUKS(I,K)=PMUKS1(I,K)*P+PMUKS2(I,K)*(1.-P)
C
C      CALCULATE COVARIANCE MATRIX
C
       DO 2 I=1,N
C
       TMP1(I,1)=PMUKS1(I,K)
     2 S2K(I,1)=MUKS(I,K)
C
       CALL SUB(TMP1,S2K,N,1,TMP1)
       CALL TRANS(TMP1,N,1,TMP2)
       CALL PROD(TMP1,TMP2,N,1,N,TMP1)
C
       DO 3 I=1,N
C
       DO 3 J=1,N
     3 TMP2(I,J)=PKS1(I,J,K)
C
       CALL ADD(TMP2,TMP1,N,N,TMP1)
C
       DO 4 I=1,N
C
       DO 4 J=1,N
     4 S1K(I,J)=TMP1(I,J)*P
C
        DO 5 I=1,N
     5 TMP1(I,1)=PMUKS2(I,K)
C
       CALL SUB(TMP1,S2K,N,1,TMP1)
       CALL TRANS(TMP1,N,1,TMP2)
       CALL PROD(TMP1,TMP2,N,1,N,TMP1)
C
       DO 6 I=1,N
```

```
C
      DO 6 J=1,N
    6 TMP2(I,J)=PKS2(I,J,K)
C
      CALL ADD(TMP2,TMP1,N,N,TMP1)
C
      DO 7 I=1,N
C
      DO 7 J=1,N
    7 S2K(I,J)=TMP1(I,J)*(1.-P)
C
      DO 8 I=1,N
C
      DO 8 J=1,N
    8 PKS(I,J,K)=S1K(I,J)+S2K(I,J)
C
    9 CONTINUE
      RETURN
      END
      SUBROUTINE QON
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 MOK,M1K,M2K,MUK,MUKS
      COMMON GKS(9,9,60),PKS(9,9,60),XKS(9,60),MUKS(9,60),SOK(9,9),
     1S1K(9,9),S2K(9,9),S3K(9,9),PHI(9,9),C(9,9),H(9,9),EID(9,9),
     2A1K(9,9),A2K(9,9),GK(9,9),MOK(9,9),M1K(9,9),M2K(9,9),SOKM1(9,9),
     3S1KM1(9,9),RK(9,9),SOKM2(9,9),RKM1(9,9),RKM2(9,9),S3KM1(9,9),
     4TMP1(9,9),TMP2(9,9),PK(9,9),XK(9),VTMP1(9),XHIT(9),VTMP2(9),
     5CON(9),VTMP3(9),MUK(9),XKM1(9),XKM2(9),DK(9),N,L,NA,K,ND,NSAM,
     6IP,IP1,IH,IQ,PHK(9,9,60),HK(9,9,60),QK(9,9,60),PMUKS1(9,60),
     7PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)
C
      DELT=100.D0
      SIGVS=31.7D-06
      SIGTS=3.04D-06
      SIGFS=.25D-06
      A=-1.D0
      DELT1=DELT**2
C
      DO 1 K=1,NSAM
C
      QK(1,1,K)=(XKS(2,K)**2)*DELT1*SIGTS
      QK(1,2,K)=-(XKS(1,K)*XKS(2,K)*DELT1*SIGTS)
      QK(1,3,K)=0.D0
C
      QK(1,4,K)=A*XKS(2,K)*DELT1*SIGTS
      QK(1,5,K)=0.D0
      QK(2,2,K)=((DELT1**2)/4.D0)*SIGVS+(XKS(1,K)**2)*DELT1*SIGTS
      QK(2,3,K)=((DELT**3)/2.D0)*SIGVS
```

```
      QK(2,4,K)=-(A*XKS(1,K)*DELT1*SIGTS)
      QK(2,5,K)=0.D0
      QK(3,3,K)=DELT1*SIGVS
      QK(3,4,K)=0.D0
      QK(3,5,K)=0.D0
      QK(4,4,K)=DELT1*SIGTS
      QK(4,5,K)=0.D0
      QK(5,5,K)=DELT1*SIGFS
      QK(2,1,K)=QK(1,2,K)
      QK(3,1,K)=QK(1,3,K)
      QK(3,2,K)=QK(2,3,K)
      QK(4,1,K)=QK(1,4,K)
      QK(4,2,K)=QK(2,4,K)
      QK(4,3,K)=QK(3,4,K)
      QK(5,1,K)=QK(1,5,K)
      QK(5,2,K)=QK(2,5,K)
      QK(5,3,K)=QK(3,5,K)
      QK(5,4,K)=QK(4,5,K)
    1 CONTINUE
C
C
      RETURN
      END
```

```
C     THIS PROGRAM PERFORMS MONTE CARLO SIMULATION OF STATE ESTIMATORS  MCSP0000
C     OF WHICH THE KALMAN FILTER IS ONE EXAMPLE. THERE ARE SEVERAL      MCSP0001
C     OPTIONS AVAILABLE AS INDICATED IN THE DETAILLED COMMENTS BELOW.   MCSP0002
C     IT SHOULD BE NOTED THAT ALL COMPUTATIONS OF GAINS USING THE       MCSP0003
C     SUBROUTINE GAIN ARE PERFORMED IN DOUBLE PRECISION. THUS ALL       MCSP0004
C     ARRAYS FOR USE IN "GAIN" MUST BE PREPARED ACCORDINGLY.            MCSP0005
C                                                                       MCSP0006
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,PKKM2,G,PKK,Q,EI MCSP0007
      COMMON EI(5,5),Q(5,5),G(5,5),PKK(5,5),GAMMA(5,5),COVW(5,5),
     1TEMP(5,5),TEMP1(5,5),TEMP2(5,5),H(5,5),PKKM1(5,5),R(5,5),PHI(5,5),
     2VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),
     3GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),SIGW(5),X(5),
     4SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VTMP(5),Z(5),V(5),SIGV(5),
     5XHATZ(5),C(6),IEXT,IPHI,IH,
     6N,NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,M
      DIMENSION XP(60),YP(60),PKKM2(5,5)
C                                                                       MCSP0016
C                                                                       MCSP0017
C     N=ORDER OF SYSTEM MODEL AND FILTER (DIMENSION OF X,XHAT)          MCSP0018
C                                                                       MCSP0019
C     M=NUMBER OF MEASUREMENTS (DIMENSION OF THE VECTOR Z)              MCSP0020
C                                                                       MCSP0021
C     IN=NUMBER OF INPUT RANDOM FORCING FCNS (=DIMENSION OF W)          MCSP0022
C                                                                       MCSP0023
C     NSAM=NUMBER OF TIME SAMPLES                                       MCSP0024
C                                                                       MCSP0025
C     NENS=NUMBER OF MEMBERS IN ENSEMBLE                                MCSP0026
C                                                                       MCSP0027
      READ (5,81) N,M,IN,NSAM,NENS                                      MCSP0028
C                                                                       MCSP0029
C                                                                       MCSP0030
      READ (5,82) ND                                                    MCSP0031
C     THE VALUE OF ND READ IN MUST EQUAL THE ROW (AND COLUMN) DIMENSION MCSP0032
C     SPECIFIED FOR THE SQUARE MATRIX "TEMP1", E.G. IF TEMP1(3,3) IS    MCSP0033
C     SPECIFIED IN THE COMMON STATEMENT "ND" MUST BE EQUAL TO 3.        MCSP0034
C                                                                       MCSP0035
C                                                                       MCSP0036
C     IG=-1 -- GAINS COMPUTED OFF-LINE AND READ IN                      MCSP0037
C         0 -- GAINS COMPUTED ONLY ONCE BEFORE STARTING MONTE CARLO     MCSP0038
C         1 -- GAINS COMPUTED FOR EACH MEMBER OF ENSEMBLE               MCSP0039
C                                                                       MCSP0040
C     IFLR=0 -- R IS READ IN                                            MCSP0041
C     IFLR.NE.0 -- R IS COMPUTED ON-LINE AT EACH TIME SAMPLE            MCSP0042
C                                                                       MCSP0043
C     IEST=0 -- STANDARD KALMAN FILTER EQUATION IS USED                 MCSP0044
C     IEST.NE.0 -- STD. KALMAN FILTER EQ. NOT USED                      MCSP0045
C                                                                       MCSP0046
C     ITRK= 0 -- SEVERAL TRACKS GENERATED FROM STD LINEAR EQS           MCSP0047
```

```
C     ITRK= 1 -- ONLY ONE TRACK IS USED                                 MCSP0048
C      ITRK.NE.0OR1 -- SEVERAL TRACKS GENERATED BUT NOT FROM STD.       MCSP0049
C                      LINEAR DIFFERENCE EQUATIONS                      MCSP0050
C                                                                       MCSP0051
C     ISTAT=0 -- MEAN OF TRACK,MEAN & VARIANCE OF EST. ERROR COMPUTED   MCSP0052
C     ISTAT.NE.0 -- SAME AS ISTAT=0 BUT OFF-DIAGONAL TERMS IN COVARIANCEMCSP0053
C                                                                       MCSP0054
C                    MATRIX ARE ALSO COMPUTED                           MCSP0055
C                                                                       MCSP0056
C     IEXT=1  EXTENDED KALMAN FILTER WILL BE USED
C     IEXT=0  EXTENDED KALMAN FILTER WILL NOT BE USED
      READ (5,83) IG,IFLR,IEST,ITRK,ISTAT,IQ,ITRO                       MCSP0057
      READ (5,82) IEXT
C                                                                       MCSP0058
C     IQ=-1 -- Q IS COMPUTED ON-LINE AT EACH SAMPLE BY "QON"            MCSP0059
C         0 -- Q MATRIX IS READ IN                                      MCSP0060
C         1 -- COVARIANCE OF W READ IN AND Q IS  COMPUTED ONCE BEFORE   MCSP0061
C              MONTE CARLO BEGINS BY CALLING "QMAT"                     MCSP0062
C                                                                       MCSP0063
C     ITRO=0 -- ONLY USED IF ONE TRACK; ITRO=0 INDICATES THE            MCSP0064
C               TRACK IS READ IN (S.P.), OTHERWISE THE TRACK            MCSP0065
C               IS GENERATED BY A SINGLE CALL OF SUBROUTINE             MCSP0066
C               TRACK.                                                  MCSP0067
C                                                                       MCSP0068
C     IPRT=0 -- SOME OR ALL OUTPUT DATA IS PRINTED                      MCSP0069
C                                                                       MCSP0070
C     IPLT=0 -- SOME OR ALL OUTPUT DATA IS PLOTTED                      MCSP0071
      READ (5,84) IPRT,IPLT                                             MCSP0072
C                                                                       MCSP0073
C     IGPLT,ITHVPL,IMTPLT,ISMPLT,ISVPLT ARE PLOTTING INDICATORS         MCSP0074
C     IF THE INDICATOR IS EQUAL TO 1 PLOTS ARE GENERATED. THE INDICATORSMCSP0075
C     CORRESPOND TO THE FOLLOWING QUANTITIES                            MCSP0076
C     IGPLT(GAIN VS K),ITHVPL (THEORETICAL COVARIANCE MATRIX VS K),     MCSP0077
C     IMTPLT (SAMPLE MEAN OF TRACK=TRACK IF ONLY ONE TRACK),            MCSP0078
C     ISMPLT (SAMPLE MEAN OF EST. ERROR VS K), ISVPLT (SAMPLE VARIANCE  MCSP0079
C     OF EST. ERROR VS K)                                               MCSP0080
      IF (IPLT.EQ.0) READ (5,85) IGPLT,ITHVPL,IMTPLT,ISMPLT,ISVPLT      MCSP0081
C                                                                       MCSP0082
C     IPHI=0 -- PHI MATRIX TO BE READ IN (IN DOUBLE PREC)               MCSP0083
C     IPHI.NE.0 -- PHI MATRIX NOT TO BE READ                            MCSP0084
C                                                                       MCSP0085
C     IH=0 -- H MATRIX TO BE READ IN (IN D.P.)                          MCSP0086
C                                                                       MCSP0087
C     IR=0 -- R MATRIX IS READ IN                                       MCSP0088
C                                                                       MCSP0089
C     IPKKM1=0 -- THE STARTING VALUE FOR THE MATRIX P(K/K-1) IS TO      MCSP0090
C                 BE READ IN (D.P.)                                     MCSP0091
C                                                                       MCSP0092
```

```
C     IGAM=0 -- GAMMA MATRIX TO BE READ IN (D.P.)                         MCSP0093
C                                                                         MCSP0094
C     ISIGV=0 -- STD. DEVIATIONS OF MEASUREMENT NOISE TO BE READ IN (SP)MCSP0095
C                                                                         MCSP0096
C     ISIGW=0 -- STD. DEVIATIONS OF RANDOM INPUTS TO BE READ IN (SP)      MCSP0097
C                                                                         MCSP0098
C     IXHZ=0 -- INITIAL VALUE OF THE PREDICTED VALUE XH(0/-1) TO          MCSP0099
C               BE READ IN (S.P.)                                         MCSP0100
C                                                                         MCSP0101
C     IC=1 -- ONE INITIAL CONDITION VALUE USED FOR THE STATE              MCSP0102
C             OTHERWISE THE VALUE OF X(0) IS GENERATED BY A RANDOM        MCSP0103
C             NIMBER GENERATOR CALLED BY "XZERO"                          MCSP0104
C                                                                         MCSP0105
C                                                                         MCSP0106
      READ (5,86) IPHI,IH,IR,IPKKM1,IGAM,ISIGV,ISIGW,IXHZ,IC              MCSP0107
      CALL OVFLOW                                                         MCSP0108
      IW = 6395217                                                        MCSP0109
      IV = 1936748                                                        MCSP0110
      IXZ = 135769                                                        MCSP0111
C                                                                         MCSP0112
C                                                                         MCSP0113
C     THE FOLLOWING SECTION PRINTS OUT A DESCRIPTION OF THE RUN AS        MCSP0114
C     SPECIFIED BY THE USER'S FLAGS                                       MCSP0115
C                                                                         MCSP0116
C                                                                         MCSP0117
      WRITE (6,156)                                                       MCSP0118
      WRITE (6,87)                                                        MCSP0119
      IF (IG.EQ.0) GO TO 1                                                MCSP0120
      IF (IG.EQ.1) GO TO 2                                                MCSP0121
      WRITE (6,88)                                                        MCSP0122
      GO TO 3                                                             MCSP0123
    1 WRITE (6,89)                                                        MCSP0124
      GO TO 3                                                             MCSP0125
    2 WRITE (6,90)                                                        MCSP0126
    3 IF (IEST.EQ.0) GO TO 4                                              MCSP0127
      WRITE (6,91)                                                        MCSP0128
      GO TO 5                                                             MCSP0129
    4 WRITE (6,92)                                                        MCSP0130
    5 IF (ITRK.EQ.0) GO TO 8                                              MCSP0131
      IF (ITRK.EQ.-1) GO TO 7                                             MCSP0132
      IF (ITRO.EQ.0) GO TO 6                                              MCSP0133
      WRITE (6,93)                                                        MCSP0134
      GO TO 9                                                             MCSP0135
    6 WRITE (6,94)                                                        MCSP0136
      GO TO 9                                                             MCSP0137
    7 WRITE (6,95)                                                        MCSP0138
      GO TO 9                                                             MCSP0139
    8 WRITE (6,96)                                                        MCSP0140
```

```
 9 IF (ISTAT.EQ.0) GO TO 10
   WRITE (6,97)
   GO TO 11
10 WRITE (6,98)
11 IF (IQ.EQ.0) GO TO 13
   IF (IQ.EQ.1) GO TO 12
   WRITE (6,99)
   GO TO 14
12 WRITE (6,100)
   GO TO 14
13 WRITE (6,101)
14 IF (IFLR.EQ.0) GO TO 15
   WRITE (6,102)
   GO TO 16
15 WRITE (6,103)
16 WRITE (6,104)
   IF (IPHI.EQ.0) WRITE (6,105)
   IF (IH.EQ.0) WRITE (6,106)
   IF (IFLR.EQ.0) WRITE (6,107)
   IF (IQ.EQ.0) GO TO 17
   IF (IQ.EQ.-1) GO TO 18
   WRITE (6,108)
   GO TO 18
17 WRITE (6,109)
18 IF (IGAM.EQ.0) WRITE (6,110)
   IF (ISIGV.EQ.0) WRITE (6,111)
   IF (ISIGW.EQ.0) WRITE (6,112)
   IF (IXHZ.EQ.0) WRITE (6,113)
   IF (IC.NE.1) WRITE (6,114)
   IF (IPKKM1.EQ.0) WRITE (6,115)
C
C
C  THE FOLLOWING SECTION PRINTS OUT A DESCRIPTION OF THE OUTPUT
C  DATA CALLED FOR
C
C
   WRITE (6,116)
   IF (IPRT.NE.0) GO TO 19
   WRITE (6,117)
   IF (IG.EQ.0) WRITE (6,118)
   WRITE (6,119)
   IF (ISTAT.NE.0) WRITE (6,120)
   GO TO 20
19 WRITE (6,121)
20 IF (IPLT.NE.0) GO TO 21
   WRITE (6,122)
   IF (IGPLT.EQ.1) WRITE (6,123)
   IF (ITHVPL.EQ.1) WRITE (6,124)
```

```
      IF (IMTPLT.EQ.1) WRITE (6,125)
      IF (ISMPLT.EQ.1) WRITE (6,126)
      IF (ISVPLT.EQ.1) WRITE (6,127)
      GO TO 22
   21 WRITE (6,128)
   22 CONTINUE
      WRITE (6,156)
      WRITE (6,129)
      WRITE (6,130) N,M,IN,NSAM,NENS,ND
C
C
C     THE FOLLOWING SECTION READS THE SPECIFIED INPUT MATRICES
C
      IF (IPHI.NE.0) GO TO 24
      CALL MREAD (PHI,N,N)
C
      DO 23 I=1,N
C
      DO 23 J=1,N
   23 PHIS(I,J) = PHI(I,J)
C
      WRITE (6,131)
      CALL MWRITE (PHI,N,N)
C
C
   24 IF (IH.NE.0) GO TO 26
      CALL MREAD (H,M,N)
C
      DO 25 I=1,M
C
      DO 25 J=1,N
   25 HS(I,J) = H(I,J)
C
      WRITE (6,132)
      CALL MWRITE (H,M,N)
C
C
   26 IF (IR.NE.0) GO TO 27
      CALL MREAD (R,M,M)
      WRITE (6,133)
      CALL MWRITE (R,M,M)
C
C
   27 IF (IQ.NE.1) GO TO 28
      CALL MREAD (COVW,IN,IN)
      WRITE (6,134)
      CALL MWRITE (COVW,IN,IN)
      GO TO 29
```

```
   28 IF (IQ.NE.0) GO TO 29                                         MCSP0237
      CALL MREAD (Q,N,N)                                            MCSP0238
      WRITE (6,135)                                                 MCSP0239
      CALL MWRITE (Q,N,N)                                           MCSP0240
C                                                                   MCSP0241
C                                                                   MCSP0242
   29 IF (IGAM.NE.0) GO TO 31                                       MCSP0243
      CALL MREAD (GAMMA,N,IN)                                       MCSP0244
C                                                                   MCSP0245
      DO 30 I=1,N                                                   MCSP0246
C                                                                   MCSP0247
      DO 30 J=1,IN                                                  MCSP0248
   30 GAMMAS(I,J) = GAMMA(I,J)                                      MCSP0249
C                                                                   MCSP0250
      WRITE (6,136)                                                 MCSP0251
      CALL MWRITE (GAMMA,N,IN)                                      MCSP0252
C                                                                   MCSP0253
C                                                                   MCSP0254
   31 IF (IPKKM1.NE.0) GO TO 32                                     MCSP0255
      CALL MREAD (PKKM2,N,N)                                        MCSP0256
      WRITE (6,137)                                                 MCSP0257
      CALL MWRITE (PKKM2,N,N)                                       MCSP0258
      DO 165 I=1,N
      DO 165 J=1,N
      PKKM1(I,J)=PKKM2(I,J)
  165 CONTINUE
C                                                                   MCSP0259
C                                                                   MCSP0260
   32 IF (ISIGV.NE.0) GO TO 33                                      MCSP0261
      CALL VREAD (SIGV,M)                                           MCSP0262
      WRITE (6,138)                                                 MCSP0263
      CALL VWRITE (SIGV,M)                                          MCSP0264
C                                                                   MCSP0265
C                                                                   MCSP0266
   33 IF (ISIGW.NE.0) GO TO 34                                      MCSP0267
      CALL VREAD (SIGW,IN)                                          MCSP0268
      WRITE (6,139)                                                 MCSP0269
      CALL VWRITE (SIGW,IN)                                         MCSP0270
C                                                                   MCSP0271
C                                                                   MCSP0272
   34 IF (IXHZ.NE.0) GO TO 35                                       MCSP0273
      CALL VREAD (XHATZ,N)                                          MCSP0274
      WRITE (6,140)                                                 MCSP0275
      CALL VWRITE (XHATZ,N)                                         MCSP0276
C                                                                   MCSP0277
C                                                                   MCSP0278
   35 IF (IC.EQ.1) GO TO 36                                         MCSP0279
C     IC.NE.1 MEANS THAT MEANS AND STD. DEVIATIONS OF THE INITIAL STATE MCSP0280
```

```
C     VALUE MUST BE READ IN. OTHERWISE NOT READ IN.                      MCSP0281
      CALL VREAD (XZMEAN,N)                                              MCSP0282
      WRITE (6,141)                                                      MCSP0283
      CALL VWRITE (XZMEAN,N)                                             MCSP0284
C                                                                        MCSP0285
C                                                                        MCSP0286
      CALL VREAD (SIGXZ,N)                                               MCSP0287
      WRITE (6,142)                                                      MCSP0288
      CALL VWRITE (SIGXZ,N)                                              MCSP0289
      GO TO 40                                                           MCSP0290
C                                                                        MCSP0291
C                                                                        MCSP0292
   36 READ (5,144) (XS(I,1),I=1,N)                                       MCSP0293
C     INITIAL CONDITION HAS BEEN READ                                    MCSP0294
      WRITE (6,143)                                                      MCSP0295
      WRITE (6,146) (XS(I,1),I=1,N)                                      MCSP0296
      IF (ITRK.NE.1) GO TO 40                                            MCSP0297
      IF (ITRO.NE.0) GO TO 38                                            MCSP0298
C                                                                        MCSP0299
      DO 37 K=2,NSAM                                                     MCSP0300
   37 READ (5,144) (XS(I,K),I=1,N)                                       MCSP0301
C                                                                        MCSP0302
      GO TO 39                                                           MCSP0303
   38 CALL TRACK                                                         MCSP0304
C     IF TRACK CALLED HERE IT SHOULD BE WRITTEN TO GENERATE AND          MCSP0305
C     STORE THE TRACK IN XS(N,K) FOR K=2(NOTE),NSAM                      MCSP0306
C                                                                        MCSP0307
   39 WRITE (6,145)                                                      MCSP0308
      WRITE (6,146) (XS(I,1),I=1,N)                                      MCSP0309
      WRITE (6,146) (XS(I,NSAM),I=1,N)                                   MCSP0310
   40 CONTINUE                                                           MCSP0311
C                                                                        MCSP0312
C                                                                        MCSP0313
C     THE FOLLOWING SECTION PREPARES FOR THE MONTE CARLO LOOP            MCSP0314
C                                                                        MCSP0315
C     FORM NXN IDENTITY MATRIX IN DOUBLE PRECISION                       MCSP0316
C                                                                        MCSP0317
      DO 41 I=1,N                                                        MCSP0318
C                                                                        MCSP0319
      DO 41 J=1,N                                                        MCSP0320
      EI(I,J) = 0.D0                                                     MCSP0321
   41 IF (I.EQ.J) EI(I,J)=1.D0                                           MCSP0322
C                                                                        MCSP0323
C     GIVEN THE MATRIX GAMMA AND THE COVARIANCE OF W COMPUTE Q           MCSP0324
C     USING DOUBLE PRECISION ARITHMETIC                                  MCSP0325
      IF (IQ.NE.1) GO TO 42                                              MCSP0326
      CALL QMAT                                                          MCSP0327
      WRITE (6,135)                                                      MCSP0328
```

```
      CALL MWRITE (Q,N,N)                                               MCSP0329
   42 IF (IG.NE.-1) GO TO 44                                            MCSP0330
C                                                                       MCSP0331
      DO 43 K=1,NSAM                                                    MCSP0332
C                                                                       MCSP0333
      DO 43 I=1,N                                                       MCSP0334
   43 READ (5,144) (GKS(I,J,K),J=1,M)                                   MCSP0335
C                                                                       MCSP0336
      GO TO 47                                                          MCSP0337
   44 IF (IG.NE.0) GO TO 47                                             MCSP0338
C                                                                       MCSP0339
      DO 46 K=1,NSAM                                                    MCSP0340
      CALL GAIN                                                         MCSP0341
C                                                                       MCSP0342
      DO 46 I=1,N                                                       MCSP0343
C                                                                       MCSP0344
      DO 45 L=1,N                                                       MCSP0345
   45 PKKS(I,L,K) = PKK(I,L)                                            MCSP0346
C                                                                       MCSP0347
C                                                                       MCSP0348
      DO 46 J=1,M                                                       MCSP0349
   46 GKS(I,J,K) = G(I,J)                                               MCSP0350
C                                                                       MCSP0351
   47 CONTINUE                                                          MCSP0352
C     IF GAINS WERE TO BE READ IN (IG=-1) OR COMPUTED ONLY              MCSP0353
C     ONCE(IG=0), THIS HAS NOW BEEN DONE                                MCSP0354
C                                                                       MCSP0355
C                                                                       MCSP0356
C     SET UP ARRAYS FOR COMPUTING STATISTICS                            MCSP0357
C                                                                       MCSP0358
      DO 48 K=1,NSAM                                                    MCSP0359
C                                                                       MCSP0360
      DO 48 J=1,N                                                       MCSP0361
      XM(J,K) = 0.                                                      MCSP0362
      ERR(J,K) = 0.                                                     MCSP0363
C                                                                       MCSP0364
      DO 48 L=1,N                                                       MCSP0365
   48 VAR(J,L,K) = 0.                                                   MCSP0366
C                                                                       MCSP0367
C                                                                       MCSP0368
C                                                                       MCSP0369
C     BEGIN MAIN ITERATION LOOP HERE                                    MCSP0370
C                                                                       MCSP0371
      DO 54 ITER=1,NENS                                                 MCSP0372
      IF (IC.EQ.1) GO TO 49                                             MCSP0373
      CALL XZERO                                                        MCSP0374
C                                                                       MCSP0375
   49 DO 50 I=1,N                                                       MCSP0376
```

```
   50 XHKKM1(I) = XHATZ(I)
C
C
      DO 164 I=1,N
      DO 164 J=1,N
      PKKM1(I,J)=PKKM2(I,J)
  164 CONTINUE
C
C
      DO 54 K=1,NSAM
C     FORM NOISY MEASUREMENT FROM TRUE STATE VALUE
C
C
      DO 51 I=1,N
   51 X(I) = XS(I,K)
C
      CALL MEAS
C     GAIN IS NOT TO BE COMPUTED ON-LINE IF IG.NE.1
      IF (IG.NE.1) GO TO 53
C
C     Q IS TO BE COMPJTED ON-LINE IF IFLQ.NE.0
C     R IS TO BE COMPUTED ON-LINE IF IFLR.NE.0
C
      IF (IFLR.NE.0) CALL RON
      CALL GAIN
C
      DO 52 I=1,N
C
      DO 52 J=1,M
   52 GKS(I,J,K) = G(I,J)
C
C
C     UPDATE THE STATE ESTIMATE
C
   53 IF (IEXT.EQ.0) CALL ESTIM
C
C     UPDATE RUNNING SUMS USED IN COMPUTING STATISTICS
      CALL STAT
      IF (K.EQ.NSAM) GO TO 54
C
C     UPDATE TRACK BY COMPUTING X(K+1)
      IF (ITRK.NE.1) CALL TRACK
   54 CONTINUE
C
C
C     DIVIDE RUNNING SUMS COMPUTED BY SUBROUTINE STAT BY ENSEMBLE
C     SIZE TO COMPUTE STATISTICS
      ENS = NENS
```

```
C                                                                       MCSP0420
      DO 56 K=1,NSAM                                                    MCSP0421
C                                                                       MCSP0422
      DO 56 J=1,N                                                       MCSP0423
      IF (ITRK.EQ.1) GO TO 55                                           MCSP0424
      XM(J,K) = XM(J,K)/ENS                                             MCSP0425
   55 ERR(J,K) = ERR(J,K)/ENS                                           MCSP0426
   56 VAR(J,J,K) = VAR(J,J,K)/ENS-ERR(J,K)**2                           MCSP0427
C                                                                       MCSP0428
      IF (ISTAT.EQ.0) GO TO 58                                          MCSP0429
C                                                                       MCSP0430
C     COMPUTE OFF-DIAGONAL TERMS IN COVARIANCE OF ESTIMATION            MCSP0431
C     ERROR MATRIX IF ISTAT.NE.0                                        MCSP0432
C                                                                       MCSP0433
      DO 57 K=1,NSAM                                                    MCSP0434
C                                                                       MCSP0435
      DO 57 L=2,N                                                       MCSP0436
      LM1 = L-1                                                         MCSP0437
C                                                                       MCSP0438
      DO 57 J=1,LM1                                                     MCSP0439
   57 VAR(L,J,K) = VAR(L,J,K)/ENS-ERR(L,K)*ERR(J,K)                     MCSP0440
C                                                                       MCSP0441
   58 CONTINUE                                                          MCSP0442
C                                                                       MCSP0443
C                                                                       MCSP0444
      IF (IPRT.NE.0) GO TO 64                                           MCSP0445
      WRITE (6,147)                                                     MCSP0446
C     WRITE GAINS, THEORETICAL COVARIANCES OF ESTIMATION ERROR          MCSP0447
C     IF ONE SET OF GAINS HAS BEEN USED.                                MCSP0448
      WRITE (6,148)                                                     MCSP0450
C                                                                       MCSP0451
      DO 59 K=1,NSAM                                                    MCSP0452
      WRITE (6,149) K                                                   MCSP0453
C                                                                       MCSP0454
      DO 59 I=1,N                                                       MCSP0455
   59 WRITE (6,146) (GKS(I,J,K),J=1,M)                                  MCSP0456
C                                                                       MCSP0457
      WRITE (6,150)                                                     MCSP0458
C                                                                       MCSP0459
      DO 60 K=1,NSAM                                                    MCSP0460
      WRITE (6,151) K                                                   MCSP0461
C                                                                       MCSP0462
      DO 60 I=1,N                                                       MCSP0463
   60 WRITE (6,146) (PKKS(I,J,K),J=1,N)                                 MCSP0464
C                                                                       MCSP0465
C                                                                       MCSP0466
C                                                                       MCSP0467
   61 WRITE (6,156)                                                     MCSP0468
```

```
      WRITE (6,152)
      WRITE (6,153)
C
      DO 62 K=1,NSAM
      WRITE (6,155)
C
      DO 62 I=1,N
   62 WRITE (6,154) K,I,XM(I,K),ERR(I,K),VAR(I,I,K)
C
      WRITE (6,156)
      IF (ISTAT.EQ.0) GO TO 64
      WRITE (6,157)
C
      DO 63 K=1,NSAM
      WRITE (6,158) K
C
      DO 63 I=1,N
   63 WRITE (6,146) (VAR(I,L,K),L=1,I)
C
      WRITE (6,156)
   64 IF (IPLT.NE.0) GO TO 80
C
      DO 65 K=1,NSAM
   65 XP(K) = K
C
      IF (IGPLT.NE.1) GO TO 68
C
      DO 67 I=1,N
C
      DO 67 J=1,M
C
      DO 66 K=1,NSAM
   66 YP(K) = GKS(I,J,K)
C
      WRITE (6,156)
      CALL PLOTP (XP,YP,NSAM,0)
   67 WRITE (6,159) I,J
C
   68 IF (ITHVPL.NE.1) GO TO 71
C
      DO 70 I=1,N
C
      DO 69 K=1,NSAM
   69 YP(K) = PKKS(I,I,K)
C
      WRITE (6,156)
      CALL PLOTP (XP,YP,NSAM,0)
   70 WRITE (6,160) I,I
```

```
C
   71 IF (IMTPLT.NE.1) GO TO 74
C
      DO 73 I=1,N
C
      DO 72 K=1,NSAM
   72 YP(K) = XM(I,K)
C
      WRITE (6,156)
      CALL PLOTP (XP,YP,NSAM,0)
   73 WRITE (6,161) I
C
   74 IF (ISMPLT.NE.1) GO TO 77
C
      DO 76 I=1,N
C
      DO 75 K=1,NSAM
   75 YP(K) = ERR(I,K)
C
      WRITE (6,156)
      CALL PLOTP (XP,YP,NSAM,0)
   76 WRITE (6,162) I,I
C
   77 IF (ISVPLT.NE.1) GO TO 80
C
      DO 79 I=1,N
C
      DO 78 K=1,NSAM
   78 YP(K) = VAR(I,I,K)
C
      WRITE (6,156)
      CALL PLOTP (XP,YP,NSAM,0)
   79 WRITE (6,163) I
C
      WRITE (6,156)
   80 CONTINUE
C     ****************************************************************************************
C      THE FOLLOWING SECTION IS TESTING ONLY
C
C     ONCETESTING IS COMPLATED.
C
  500 FORMAT (I1)
C
      READ (5,500) IT
      IF (IT.NE.1) GO TO 501
      DO 517 K=1,NSAM
  517 READ(5,520) (XS(I,K),I=1,N)
  520 FORMAT (4(1PE20.12))
```

```
C
      DO 502 K=1,NSAM
      DO 502 IK=1,N
  502 READ (5,520) (PKKS(IK,J,K),J=1,N)
C
      WRITE (6,504)
  504 FORMAT (2X,'THE MEANS OF EST. ERROR CALC. FROM EVAL ARE',/)
C
      DO 518 K=1,NSAM
  518 WRITE (6,505) K,(XS(I,K),I=1,N)
  505 FORMAT (2X,'K=',I3,5X,6(1PE12.5),/)
C
      DO 503 K=1,NSAM
      DO 503 IK=1,N
  503 WRITE (6,514) (PKKS(IK,J,K),J=1,N)
C
C     COMPUTE RMS ERRORS OF STATISTICS
C
      DO 509 I=1,N
      X(I)=0.
      DO 509 J=1,N
  509 GK(I,J)=0.
C
      DO 510 I=1,N
      DO 510 J=1,I
      DO 510 K=1,NSAM
  510 GK(I,J)=GK(I,J)+VAR(I,J,K)**2+PKKS(I,J,K)**2-2.*VAR(I,J,K)*PKKS
     .(I,J,K)
C
      DO 521 I=1,N
      DO 521 K=1,NSAM
  521 X(I)=X(I)+ERR(I,K)**2+XS(I,K)**2-2.*ERR(I,K)*XS(I,K)
C
      ENSAM=NSAM
      DO 511 I=1,N
      X(I)=SQRT(X(I)/ENSAM)
      DO 511 J=1,I
  511 GK(I,J)=SQRT(GK(I,J)/ENSAM)
C
      WRITE (6,512) (X(I),I=1,N)
      WRITE (6,515)
  515 FORMAT (5X,'THE RMS DEVIATIONS OF EST ERROR VARIANCES ARE',/)
C
      DO 513 I=1,N
  513 WRITE (6,514) (GK(I,J),J=1,I)
  512 FORMAT (5X,'RMS DEVIATIONS OF EST ERROR MEANS ARE',/,9(2X,1PE12.5)
     .,/)
  514 FORMAT (9(2X,1PE12.5),/)
```

```
  501 CONTINUE
      STOP                                                              MCSP0553
C                                                                       MCSP0554
   81 FORMAT (5(I10))                                                   MCSP0555
   82 FORMAT (I2)                                                       MCSP0556
   83 FORMAT (7(I10))                                                   MCSP0557
   84 FORMAT (2(I5))                                                    MCSP0558
   85 FORMAT (5(I10))                                                   MCSP0559
   86 FORMAT (9I5)                                                      MCSP0560
   87 FORMAT (20X,'DESCRIPTION OF RUN',///)                             MCSP0561
   88 FORMAT (10X,'GAINS COMPUTED OFF-LINE AND READ IN',/)              MCSP0562
   89 FORMAT (10X,'GAINS COMPUTED ONCE IN "GAIN" BEFORE STARTING MONTE CMCSP0563
     1ARLO',/)                                                          MCSP0564
   90 FORMAT (10X,'GAINS COMPUTED FOR EACH MEMBER OF ENSEMBLE',/)       MCSP0565
   91 FORMAT (10X,'THE STANDARD LINEAR EQS. DO NOT CHARACTERIZE THE FILTMCSP0566
     1ER',/)                                                            MCSP0567
   92 FORMAT (10X,'THE STD. KALMAN EQS. CHARACTERIZE THE LINEAR FILTER',MCSP0568
     1/)                                                                MCSP0569
   93 FORMAT (10X,'ONLY ONE TRACK IS USED AND IT IS GENERATED BY SUBROUTMCSP0570
     1INE TRACK',/)                                                     MCSP0571
   94 FORMAT (10X,'ONLY ONE TRACK IS USED AND IT IS READ IN',/)         MCSP0572
   95 FORMAT (10X,'SEVERAL TRACKS USED BUT NOT GENERATED FROM STD. LINEAMCSP0573
     1R DIFFERENCE EQS.',/)                                             MCSP0574
   96 FORMAT (10X,'SEVERAL TRACKS GENERATED BY USING THE STD. LINEAR DIFMCSP0575
     1FERENCE EQS.',/)                                                  MCSP0576
   97 FORMAT (10X,'MEAN OF TRACK, MEAN OF EST. ERROR AND COVARIANCE OF EMCSP0577
     1ST. ERROR ARE COMPUTED',/)                                        MCSP0578
   98 FORMAT (10X,'MEAN OF TRACK, MEAN AND VARRIANCES OF EST. ERROR ARE MCSP0579
     1COMPUTED',/)                                                      MCSP0580
   99 FORMAT (10X,'THE Q MATRIX IS COMPUTED ON-LINE AT EACH SAMPLE BY "QMCSP0581
     1ON"',/)                                                           MCSP0582
  100 FORMAT (10X,'THE COVARIANCE OF W IS READ IN AND Q IS COMPUTED BY "MCSP0583
     1QMAT" BEFORE STARTING MONTE CARLO',/)                             MCSP0584
  101 FORMAT (10X,'THE Q MATRIX IS READ IN',/)                          MCSP0585
  102 FORMAT (10X,'R IS COMPUTED ON-LINE AT EACH SAMPLE BY "RON"',/)    MCSP0586
  103 FORMAT (10X,'R IS READ IN',/)                                     MCSP0587
  104 FORMAT (////,20X,'INPUT DATA CALLED FOR',///)                     MCSP0588
  105 FORMAT (10X,'PHI MATRIX',/)                                       MCSP0589
  106 FORMAT (10X,'H MATRIX',/)                                         MCSP0590
  107 FORMAT (10X,'R MATRIX',/)                                         MCSP0591
  108 FORMAT (10X,'COVARIANCE OF W',/)                                  MCSP0592
  109 FORMAT (10X,'Q MATRIX',/)                                         MCSP0593
  110 FORMAT (10X,'GAMMA MATRIX',/)                                     MCSP0594
  111 FORMAT (10X,'STANDARD DEVIATIONS OF MEASUREMENT NOISE',/)         MCSP0595
  112 FORMAT (10X,'STANDARD DEVIATIONS OF INPUT FORCING W',/)           MCSP0596
  113 FORMAT (10X,'XHAT(0/-1)',/)                                       MCSP0597
  114 FORMAT (10X,'MEANS AND VARIANCES OF X(0)',/)                      MCSP0598
  115 FORMAT (10X,'P(0/-1)',/)                                          MCSP0599
```

```
116 FORMAT (////,20X,'OUTPUT CALLED FOR',//)                             MCSP0600
117 FORMAT (10X,'PRINTED OUTPUT OF THE FOLLOWING DATA',/)                MCSP0601
118 FORMAT (15X,'GAIN MATRICES AND THEORETICAL COVARIANCE OF EST. ERROMCSP0602
   1R MATRICES',/)                                                       MCSP0603
119 FORMAT (15X,'SAMPLE MEANS OF TRACK AND ESTIMATION ERROR,SAMPLE VARMCSP0604
   1IANCES OF EST. ERROR',/)                                             MCSP0605
120 FORMAT (15X,'COVARIANCE OF ESTIMATION ERROR MATRIX',/)               MCSP0606
121 FORMAT (10X,'NO PRINTED OUTPUT CALLED FOR',//)                       MCSP0607
122 FORMAT (10X,'THE FOLLOWING PLOTS ARE CALLED FOR',/)                  MCSP0608
123 FORMAT (15X,'G(K) VS. K',/)                                          MCSP0609
124 FORMAT (15X,'P(K/K) THEORETICAL VS. K',/)                            MCSP0610
125 FORMAT (15X,'MEAN OF TRACK VS. K',/)                                 MCSP0611
126 FORMAT (15X,'SAMPLE MEANS OF ESTIMATION ERROR VS. K',/)              MCSP0612
127 FORMAT (15X,'SAMPLE VARIANCES OF ESTIMATION ERROR VS. K',/)          MCSP0613
128 FORMAT (10X,'NO PLOTS CALLED FOR',/)                                 MCSP0614
129 FORMAT (20X,'INPUT DATA',///)                                        MCSP0615
130 FORMAT (4X,'N=',I2,4X,'M=',I2,4X'IN=',I2,4X'NSAM=',I3,4X,'NENS=',IMCSP0616
   15,4X'ND=',I2,//)                                                     MCSP0617
131 FORMAT (//,10X,'THE PHI MATRIX IS',/)                                MCSP0618
132 FORMAT (//,10X,'THE H MATRIX IS',/)                                  MCSP0619
133 FORMAT (//,10X,'THE R MATRIX IS',/)                                  MCSP0620
134 FORMAT (//,10X,'THE COVARIANCE OF W MATRIX IS',/)                    MCSP0621
135 FORMAT (//,10X,'THE Q MATRIX IS',/)                                  MCSP0622
136 FORMAT (//,10X,'THE GAMMA MATRIX IS',/)                              MCSP0623
137 FORMAT (//,10X,'THE MATRIX P(0/-1) IS',/)                            MCSP0624
138 FORMAT (//,10X,'THE STD. DEVIATIONS OF MEASUREMENT NOISE ARE',/)     MCSP0625
139 FORMAT (//,10X,'THE STD. DEVIATIONS OF INPUT FORCING W ARE',/)       MCSP0626
140 FORMAT (//,10X,'THE VECTOR XHAT(0/-1) IS',/)                         MCSP0627
141 FORMAT (//,10X,'THE MEAN OF THE VECTOR X(0) IS',/)                   MCSP0628
142 FORMAT (//,10X,'THE STANDARD DEVIATIONS OF THE VECTOR X(0) ARE',/)MCSP0629
143 FORMAT (//,10X,'THE INITIAL STATE IS',/)                             MCSP0630
144 FORMAT(8(F10.0))
145 FORMAT (//,10X,'THE FIRST AND LAST POINTS ON THE SINGLE TRACK TO BMCSP0632
   1E USED ARE',/)                                                       MCSP0633
146 FORMAT (9(2X,1PE12.5),/)                                             MCSP0634
147 FORMAT ('1',20X,'OUTPUT DATA',///)                                   MCSP0635
148 FORMAT (10X,'THE GAIN MATRICES ARE',/)                               MCSP0636
149 FORMAT (5X,'K=',I3,/,10X,'G(K)=',/)                                  MCSP0637
150 FORMAT (1X,//,10X,'THE THEORETICAL COVARIANCE MATRIX IS',/)          MCSP0638
151 FORMAT (5X,'K=',I3,/,10X,'P(K/K)=',/)                                MCSP0639
152 FORMAT (T5,'TIME',T16,'VECTOR COM-',T34,'SAMPLE MEAN',               MCSP0640
   1 T51,'SAMPLE MEAN OF',T71,'SAMPLE VARIANCE OF')                      MCSP0641
153 FORMAT (T5,'INDEX',T16,'PONENT INDEX',T34,'OF TRACK',                MCSP0642
   1 T51,'ESTIMATION ERROR',T71,'ESTIMATION ERROR')                      MCSP0643
154 FORMAT (6X,I3,13X,I1,10X,1PE14.7,2(6X,1PE14.7))                      MCSP0644
155 FORMAT (//)                                                          MCSP0645
156 FORMAT ('1')                                                         MCSP0646
157 FORMAT (10X,'THE SAMPLE COVARIANCE OF EST. ERROR MATRIX IS',//)      MCSP0647
```

```
  158 FORMAT (//,2X,'K=',I3,/)                                          MCSP0648
  159 FORMAT (12X,'G(',I1,',',I1,')  VS. K')                            MCSP0649
  160 FORMAT (12X,'PKK(',I1,',',I1,')  VS. K')                          MCSP0650
  161 FORMAT (12X,'MEAN OF X(',I1,')  VS. K')                           MCSP0651
  162 FORMAT (12X,'XHATKK(',I1,') -X(',I1,')  VS. K')                   MCSP0652
  163 FORMAT (12X,'ERROR VARIANCE(',I1,') VS.. K')                      MCSP0653
      END                                                               MCSP0654
      SUBROUTINE ESTIM                                                  MCSP0655
C     THIS SUBROUTINE UPDATES THE STATE ESTIMATE. IN THE DEFAULT        MCSP0656
C     CONDITION (IEST.EQ.0) THE STANDARD EQUATIONS                      MCSP0657
C                                                                       MCSP0658
C     XHAT(K/)=XHAT(K/K-1)+G(K)*(Z(K)-H(K)*XHAT(K/K-1))                 MCSP0659
C                                                                       MCSP0660
C     XHAT(K+1/K)=PHI*XHAT(K/K)                                         MCSP0661
C                                                                       MCSP0662
C     ARE EVALUATED                                                     MCSP0663
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI       MCSP0664
      COMMON EI(5,5),Q(5,5),G(5,5),PKK(5,5),GAMMA(5,5),COVW(5,5),
     1TEMP(5,5),TEMP1(5,5),TEMP2(5,5),H(5,5),PKKM1(5,5),R(5,5),PHI(5,5),
     2VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),
     3GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),SIGW(5),X(5),
     4SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VTMP(5),Z(5),V(5),SIGV(5),
     5XHATZ(5),C(6),IEXT,IPHI,IH,
     6N,NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,M
C     TAKE THE APPROPRIATE GAIN AND STORE IN THE ARRAY GK               MCSP0672
C                                                                       MCSP0673
      DO 1 I=1,N                                                        MCSP0674
C                                                                       MCSP0675
      DO 1 J=1,M                                                        MCSP0676
    1 GK(I,J) = GKS(I,J,K)                                              MCSP0677
C                                                                       MCSP0678
      IF (IEST.NE.0) GO TO 2                                            MCSP0679
      CALL VPROD (HS,XHKKM1,M,N,VTMP)                                   MCSP0680
      CALL VSUB (Z,VTMP,M,VTMP)                                         MCSP0681
      CALL VPROD (GK,VTMP,N,M,VTMP)                                     MCSP0682
      CALL VADD (XHKKM1,VTMP,N,XHKK)                                    MCSP0683
C                                                                       MCSP0684
C     XHAT(K/K) HAS BEEN COMPUTED AND STORED IN THE ARRAY XHKK          MCSP0685
      CALL VPROD (PHIS,XHKK,N,N,XHKKM1)                                 MCSP0686
C                                                                       MCSP0687
C     XHAT(K+1/K) HAS BEEN COMPUTED AND STORED IN THE ARRAY XHKKM1      MCSP0688
      RETURN                                                            MCSP0689
    2 CONTINUE                                                          MCSP0690
      IF(IEXT.EQ.0) GO TO 3
      DO 5 I=1,N
C
      DO 5 J=1,M
    5 GK(I,J)=G(I,J)
```

```
C
      CALL CK
      CALL VSUB(Z,C,M,VTMP)
      CALL VPROD (GK,VTMP,N,M,VTMP)
      CALL VADD (XHKKM1,VTMP,N,XHKK)
C
      CALL XPRED
      RETURN
    3 CONTINUE
C     IF STANDARD EQUATIONS ARE NOT TO BE USED, THE APPROPRIATE          MCSP0691
C     EQUATIONS MUST BE INSERTED HERE BY THE USER.                       MCSP0692
      RETURN                                                             MCSP0693
      END                                                                MCSP0694
      SUBROUTINE GAIN                                                    MCSP0695
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI        MCSP0696
      COMMON EI(5,5),Q(5,5),G(5,5),PKK(5,5),GAMMA(5,5),COVW(5,5),
     1TEMP(5,5),TEMP1(5,5),TEMP2(5,5),H(5,5), PKKM1(5,5),R(5,5),PHI(5,5),
     2VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),
     3GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),SIGW(5),X(5),
     4SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VTMP(5),Z(5),V(5),SIGV(5),
     5XHATZ(5),C(6),IEXT,IPHI,IH,
     6N,NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,M
C                                                                        MCSP0704
C     G(K) = P(K/K-1)*HT*(H*P(K/K-1)*HT + R)                             MCSP0705
      IF(IH.NE.0) CALL HKK
      CALL TRANS (H,M,N,TEMP2)                                           MCSP0706
      CALL PROD (PKKM1,TEMP2,N,N,M,TEMP)                                 MCSP0707
      CALL PROD (H,TEMP,M,N,M,TEMP1)                                     MCSP0708
      CALL ADD (TEMP1,R,M,M,TEMP1)                                       MCSP0709
      IF (M.EQ.1) GO TO 2                                                MCSP0710
      MD = ND                                                            MCSP0711
      CALL GAUSS3 (M,EPS,TEMP1,TEMP2,KER,MD)                             MCSP0712
      CALL PROD (TEMP,TEMP2,N,M,M,G)                                     MCSP0713
C                                                                        MCSP0714
C     NOTE HERE  PKK(I,J) = P(K/K)  WHERE                                MCSP0715
C     P(K/K) = (I-G(K)*H)*P(K/K-1)                                       MCSP0716
    1 IF (IEXT.EQ.0) GO TO 4
      CALL ESTIM
      IF(IPHI.NE.0) CALL AK
    4 CALL PROD (G,H,N,M,N,TEMP)                                         MCSP0717
      CALL SUB (EI,TEMP,N,N,TEMP2)                                       MCSP0718
      CALL PROD (TEMP2,PKKM1,N,N,N,PKK)                                  MCSP0719
C                                                                        MCSP0720
C     NOTE HERE  PKKM1(I,J) = P(K/K-1)  WHERE                            MCSP0721
C     P(K/<-1)= PHI*P(K-1/K-1)*PHIT + Q                                  MCSP0722
      IF(IQ.EQ.-1) CALL QON
      CALL TRANS (PHI,N,N,TEMP2)                                         MCSP0723
      CALL PROD (PK<,TEMP2,N,N,N,TEMP)                                   MCSP0724
```

```
      CALL PROD (PHI,TEMP,N,N,N,TEMP1)                                  MCSP0725
      CALL ADD (TEMP1,Q,N,N,PKKM1)                                      MCSP0726
      RETURN                                                            MCSP0727
C                                                                       MCSP0728
    2 DO 3 I=1,N                                                        MCSP0729
    3 G(I,1) = TEMP(I,1)/TEMP1(1,1)                                     MCSP0730
C                                                                       MCSP0731
      GO TO 1                                                           MCSP0732
      END                                                               MCSP0733
      SUBROUTINE QMAT                                                   MCSP0764
C                                                                       MCSP0765
C     THIS SUBROUTINE COMPUTES THE MATRIX Q FROM THE EQUATION           MCSP0766
C                                                                       MCSP0767
C        Q=GAMMA* E(W*WT) * GAMMAT                                      MCSP0768
C                                                                       MCSP0769
C     DOUBLE PRECISION ARITHMETIC IS USED                               MCSP0770
C                                                                       MCSP0771
C                                                                       MCSP0772
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI       MCSP0773
      COMMON EI(5,5),Q(5,5),G(5,5),PKK(5,5),GAMMA(5,5),COVW(5,5),
     1TEMP(5,5),TEMP1(5,5),TEMP2(5,5),H(5,5),PKKM1(5,5),R(5,5),PHI(5,5),
     2VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),
     3GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),SIGW(5),X(5),
     4SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VTMP(5),Z(5),V(5),SIGV(5),
     5XHATZ(5),C(6),IEXT,IPHI,IH,
     6N,NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,M
C                                                                       MCSP0781
C                                                                       MCSP0782
      CALL PROD (GAMMA,COVW,N,IN,IN,TEMP)                               MCSP0783
      CALL TRANS (GAMMA,N,IN,TEMP1)                                     MCSP0784
      CALL PROD (TEMP,TEMP1,N,IN,N,Q)                                   MCSP0785
      RETURN                                                            MCSP0786
      END                                                               MCSP0787
      SUBROUTINE QON                                                    MCSP0788
C                                                                       MCSP0789
C     IF Q IS TO BE COMPUTED ON-LINE (IFLQ.NE.0) IT IS DONE             MCSP0790
C     IN THIS SUBROUTINE                                                MCSP0791
C                                                                       MCSP0792
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI       MCSP0793
      COMMON EI(5,5),Q(5,5),G(5,5),PKK(5,5),GAMMA(5,5),COVW(5,5),
     1TEMP(5,5),TEMP1(5,5),TEMP2(5,5),H(5,5),PKKM1(5,5),R(5,5),PHI(5,5),
     2VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),
     3GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),SIGW(5),X(5),
     4SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VTMP(5),Z(5),V(5),SIGV(5),
     5XHATZ(5),C(6),IEXT,IPHI,IH,
     6N,NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,M
C                                                                       MCSP0801
C     THE APPROPRIATE STATEMENTS FOR COMPUTING Q ON-LINE MUST           MCSP0802
```

```
C     BE INSERTED HERE BY THE USER                                      MCSP0803
C                                                                       MCSP0804
      REAL*8 DELT,DELT1,SIGVS,SIGTS,SIGFS,A,B(5)
C
      DO 1 I=1,N
    1 B(I)=XHKK(I)
C
      DELT=100.DO
      SIGVS=31.7D-06
      SIGTS=3.04D-06
      SIGFS=.25D-06
      A=-1.DO
      DELT1=DELT**2
C
      Q(1,1)=(B(2)**2)*DELT1*SIGTS
      Q(1,2)=-(B(1)*B(2)*DELT1*SIGTS)
      Q(1,3)=0.DO
      Q(1,4)=A*B(2)*DELT1*SIGTS
      Q(1,5)=0.DO
      Q(2,2)=((DELT1**2)/4.DO)*SIGVS+(B(1)**2)*DELT1*SIGTS
      Q(2,3)=((DELT**3)/2.DO)*SIGVS
      Q(2,4)=-(A*B(1)*DELT1*SIGTS)
      Q(2,5)=0.DO
      Q(3,3)=DELT1*SIGVS
      Q(3,4)=0.DO
      Q(3,5)=0.DO
      Q(4,4)=DELT1*SIGTS
      Q(4,5)=0.DO
      Q(5,5)=DELT1*SIGFS
      Q(2,1)=Q(1,2)
      Q(3,1)=Q(1,3)
      Q(3,2)=Q(2,3)
      Q(4,1)=Q(1,4)
      Q(4,2)=Q(2,4)
      Q(4,3)=Q(3,4)
      Q(5,1)=Q(1,5)
      Q(5,2)=Q(2,5)
      Q(5,3)=Q(3,5)
      Q(5,4)=Q(4,5)
C
      RETURN                                                            MCSP0805
      END                                                               MCSP0806
      SUBROUTINE RON                                                    MCSP0807
C                                                                       MCSP0808
C     IF R IS TO BE COMPUTED ON-LINE (IFLR.NE.0) IT IS DONE             MCSP0809
C     IN THIS SUBROUTINE                                                MCSP0810
C                                                                       MCSP0811
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI       MCSP0812
```

```
      COMMON EI(5,5),Q(5,5),G(5,5),PKK(5,5),GAMMA(5,5),COVW(5,5),
     1TEMP(5,5),TEMP1(5,5),TEMP2(5,5),H(5,5),PKKM1(5,5),R(5,5),PHI(5,5),
     2VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),
     3GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),SIGW(5),X(5),
     4SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VTMP(5),Z(5),V(5),SIGV(5),
     5XHATZ(5),C(6),IEXT,IPHI,IH,
     6N,NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,M
C                                                                       MCSP0820
C     THE APPROPRIATE STATEMENTS FOR COMPUTING R ON-LINE MUST           MCSP0821
C     BE INSERTED HERE BY THE USER                                      MCSP0822
C                                                                       MCSP0823
      RETURN                                                            MCSP0824
      END                                                               MCSP0825
      SUBROUTINE STAT                                                   MCSP0826
C     THIS SUBROUTINE  COMPUTES RUNNING SUMS USED IN DETERMINING THE    MCSP0827
C     SAMPLE STATISTICS OF TRACK AND ESTIMATION ERRORS. IN THE DEFAULT  MCSP0828
C     OPTION (ISTAT.EQ.0) THE STATISTICS TO BE COMPUTED ARE MEAN OF     MCSP0829
C     TRACK, MEAN OF ESTIMATION ERROR AND VARIANCE OF ESTIMATION        MCSP0830
C     ERROR. IF(ISTAT.NE.0) THE OFF-DIAGONAL TERMS IN THE COVARIANCE OF MCSP0831
C     ESTIMATION ERROR MATRIX ARE ALSO COMPUTED.                        MCSP0832
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI       MCSP0833
      COMMON EI(5,5),Q(5,5),G(5,5),PKK(5,5),GAMMA(5,5),COVW(5,5),
     1TEMP(5,5),TEMP1(5,5),TEMP2(5,5),H(5,5),PKKM1(5,5),R(5,5),PHI(5,5),
     2VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),
     3GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),SIGW(5),X(5),
     4SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VTMP(5),Z(5),V(5),SIGV(5),
     5XHATZ(5),C(6),IEXT,IPHI,IH,
     6N,NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,M
      DIMENSION EXH(5)
      IF (ITRK.NE.1) GO TO 2                                            MCSP0842
      IF (ITER.NE.1) GO TO 4                                            MCSP0843
C                                                                       MCSP0844
      DO 1 J=1,N                                                        MCSP0845
    1 XM(J,K) = XS(J,K)                                                 MCSP0846
C                                                                       MCSP0847
      GO TO 4                                                           MCSP0848
    2 CONTINUE                                                          MCSP0849
C                                                                       MCSP0850
      DO 3 J=1,N                                                        MCSP0851
    3 XM(J,K) = XM(J,K)+XS(J,K)                                         MCSP0852
C                                                                       MCSP0853
    4 CONTINUE                                                          MCSP0854
C                                                                       MCSP0855
      DO 5 J=1,N                                                        MCSP0856
      EXH(J) = XHKK(J)-XS(J,K)                                          MCSP0857
      ERR(J,K) = ERR(J,K)+EXH(J)                                        MCSP0858
    5 VAR(J,J,K) = VAR(J,J,K)+EXH(J)**2                                 MCSP0859
C                                                                       MCSP0860
```

```
      IF (ISTAT.EQ.0) RETURN
C
      DO 6 L=2,N
      LM1 = L-1
C
      DO 6 J=1,LM1
    6 VAR(L,J,K) = VAR(L,J,K)+EXH(L)*EXH(J)
C
      RETURN
      END
      SUBROUTINE XZERO
C     THIS SUBROUTINE GENERATES THE INITIAL STATE VALUE FROM A NORMAL
C     RANDOM NUMBER GENERATOR. IT IS ASSUMED THAT THE INITIAL STATE
C     HAS COMPONENTS THAT ARE INDEPENDENT
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI
      COMMON EI(5,5),Q(5,5),G(5,5),PKK(5,5),GAMMA(5,5),COVW(5,5),
     1TEMP(5,5),TEMP1(5,5),TEMP2(5,5),H(5,5),PKKM1(5,5),R(5,5),PHI(5,5),
     2VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),
     3GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),SIGW(5),X(5),
     4SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VTMP(5),Z(5),V(5),SIGV(5),
     5XHATZ(5),C(6),IEXT,IPHI,IH,
     6N,NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,M
      CALL SNORM (IXZ,X,N)
C
      DO 1 I=1,N
    1 XS(I,1) = SIGXZ(I)*X(I)+XZMEAN(I)
C
      RETURN
      END
      SUBROUTINE ADD (A,B,N,M,C)
C     THIS SUBROUTINE ADDS THE NXM MATRICES A AND B, STORING THE
C     RESULT IN C
      REAL*8 A,B,C
      DIMENSION A(5,5),B(5,5),C(5,5)
C
      DO 1 I=1,N
C
      DO 1 J=1,M
    1 C(I,J) = A(I,J)+B(I,J)
C
      RETURN
      END
      SUBROUTINE MREAD (A,N,M)
C     THIS SUBROUTINE READS AN NXM MATRIX A ACCORDING TO THE FORMAT
C     8D10.5. THE ENTRIES IN THE FIRST ROW OF A ARE READ FIRST, THEN
C     THE ENTRIES IN THE SECOND ROW, AND SO ON.
      REAL*8 A
      DIMENSION A(5,5)
```

```
C                                                                       MCSP0972
      DO 1 I=1,N                                                        MCSP0973
    1 READ (5,2) (A(I,J),J=1,M)                                         MCSP0974
C                                                                       MCSP0975
      RETURN                                                            MCSP0976
C                                                                       MCSP0977
    2 FORMAT (8F10.0)                                                   MCSP0978
      END                                                               MCSP0979
      SUBROUTINE MWRITE (A,N,M)                                         MCSP0980
C     THIS SUBROUTINE WRITES THE ENTRIES OF THE NXM MATRIX A            MCSP0981
      REAL*8 A                                                          MCSP0982
      DIMENSION A(5,5)
C                                                                       MCSP0984
      DO 1 I=1,N                                                        MCSP0985
    1 WRITE (6,2) (A(I,J),J=1,M)                                        MCSP0986
C                                                                       MCSP0987
      RETURN                                                            MCSP0988
C                                                                       MCSP0989
    2 FORMAT (9(2X,1PE12.5))                                            MCSP0990
      END                                                               MCSP0991
      SUBROUTINE PROD (A,B,N,M,L,C)                                     MCSP0992
C     THIS SUBROUTINE COMPUTES THE MATRIX PRODUCT AB AND STORES THE     MCSP0993
C     RESULT IN C                                                       MCSP0994
C     A = NXM,    B = MXL,    C = NXL                                   MCSP0995
      REAL*8 A,B,C,T                                                    MCSP0996
      DIMENSION A(5,5),B(5,5),C(5,5),T(5,5)
C                                                                       MCSP0998
      DO 1 I=1,N                                                        MCSP0999
C                                                                       MCSP1000
      DO 1 J=1,L                                                        MCSP1001
    1 T(I,J) = 0.0                                                      MCSP1002
C                                                                       MCSP1003
C                                                                       MCSP1004
      DO 2 I=1,N                                                        MCSP1005
C                                                                       MCSP1006
      DO 2 J=1,L                                                        MCSP1007
C                                                                       MCSP1008
      DO 2 K=1,M                                                        MCSP1009
    2 T(I,J) = T(I,J)+A(I,K)*B(K,J)                                     MCSP1010
C                                                                       MCSP1011
C                                                                       MCSP1012
      DO 3 I=1,N                                                        MCSP1013
C                                                                       MCSP1014
      DO 3 J=1,L                                                        MCSP1015
    3 C(I,J) = T(I,J)                                                   MCSP1016
C                                                                       MCSP1017
      RETURN                                                            MCSP1018
      END                                                               MCSP1019
```

```
      SUBROUTINE SUB (A,B,N,M,C)                                        MCSP1020
C     THIS SUBROUTINE SUBTRACTS THE NXM MATRIX B FROM THE NXM MATRIX    MCSP1021
C     A AND STORES THE RESULT IN C                                      MCSP1022
      REAL*8 A,B,C                                                      MCSP1023
      DIMENSION A(5,5),B(5,5),C(5,5)
C                                                                       MCSP1025
      DC 1 I=1,N                                                        MCSP1026
C                                                                       MCSP1027
      DO 1 J=1,M                                                        MCSP1028
    1 C(I,J) = A(I,J)-B(I,J)                                            MCSP1029
C                                                                       MCSP1030
      RETURN                                                            MCSP1031
      END                                                               MCSP1032
      SUBROUTINE TRANS (A,N,M,C)                                        MCSP1033
C     THIS SUBROUTINE FORMS THE MATRIX TRANSPOSE OF A STORING THE       MCSP1034
C     RESULT IN C                                                       MCSP1035
C     A = NXM,    C = MXN                                               MCSP1036
      REAL*8 A,C                                                        MCSP1037
      DIMENSION A(5,5),C(5,5)
C                                                                       MCSP1039
      DO 1 I=1,N                                                        MCSP1040
C                                                                       MCSP1041
      DO 1 J=1,M                                                        MCSP1042
    1 C(J,I) = A(I,J)                                                   MCSP1043
C                                                                       MCSP1044
      RETURN                                                            MCSP1045
      END                                                               MCSP1046
      SUBROUTINE VADD (X,Y,N,Z)                                         MCSP1047
C     THIS SUBROUTINE COMPUTES THE SUM OF THE N-VECTORS X AND           MCSP1048
C     Y AND STORES THE RESULT IN THE N-VECTOR Z                         MCSP1049
C                                                                       MCSP1050
      REAL*4 X(5),Y(5),Z(5)
C                                                                       MCSP1052
      DO 1 I=1,N                                                        MCSP1053
    1 Z(I) = X(I)+Y(I)                                                  MCSP1054
C                                                                       MCSP1055
      RETURN                                                            MCSP1056
      END                                                               MCSP1057
      SUBROUTINE VPROD (A,X,M,N,Y)                                      MCSP1058
C     THIS SUBROUTINE COMPUTES THE PRODUCT OF THE MXN MATRIX            MCSP1059
C     A AND THE N-VECTOR X AND STORES THE RESULT IN THE                 MCSP1060
C     M-VECTOR Y                                                        MCSP1061
C                                                                       MCSP1062
C                                                                       MCSP1063
      REAL*4 A(5,5),X(5),Y(5),T(5)
C                                                                       MCSP1065
      DO 1 I=1,M                                                        MCSP1066
      T(I) = 0.D0                                                       MCSP1067
```

```
C                                                                       MCSP1068
      DO 1 J=1,N                                                        MCSP1069
    1 T(I) = T(I)+A(I,J)*X(J)                                           MCSP1070
C                                                                       MCSP1071
C                                                                       MCSP1072
      DO 2 I=1,M                                                        MCSP1073
    2 Y(I) = T(I)                                                       MCSP1074
C                                                                       MCSP1075
      RETURN                                                            MCSP1076
      END                                                               MCSP1077
      SUBROUTINE VREAD (V,N)                                            MCSP1078
C                                                                       MCSP1079
C     THIS SUBROUTINE READS THE N-DIMENSIONAL S.P. VECTOR V             MCSP1080
C                                                                       MCSP1081
      DIMENSION V(5)
      READ (5,1) (V(I),I=1,N)                                           MCSP1083
      RETURN                                                            MCSP1084
C                                                                       MCSP1085
    1 FORMAT (8F10.0)                                                   MCSP1086
      END                                                               MCSP1087
      SUBROUTINE VSUB (X,Y,N,Z)                                         MCSP1088
C     THIS SUBROUTINE COMPUTES THE DIFFERENCE X-Y OF THE TWO            MCSP1089
C     N-VECTORS X & Y AND STORES THE RESULT IN THE N-VECTOR Z           MCSP1090
C                                                                       MCSP1091
      REAL*4 X(5),Y(5),Z(5)
C                                                                       MCSP1093
      DO 1 I=1,N                                                        MCSP1094
    1 Z(I) = X(I)-Y(I)                                                  MCSP1095
C                                                                       MCSP1096
      RETURN                                                            MCSP1097
      END                                                               MCSP1098
      SUBROUTINE VWRITE (V,N)                                           MCSP1099
C                                                                       MCSP1100
C     THIS SUBROUTINE WRITES THE N-DIMENSIONAL S.P. VECTOR V            MCSP1101
C                                                                       MCSP1102
      DIMENSION V(5)
      WRITE (6,1) (V(I),I=1,N)                                          MCSP1104
      RETURN                                                            MCSP1105
C                                                                       MCSP1106
    1 FORMAT (9(2X,1PE12.5))                                            MCSP1107
      END                                                               MCSP1108
      SUBROUTINE MEAS
C     THIS SUBROUTINE STARTS WITH THE TRUE STATE VALUE XS               ***00910
C     AND ADDS ZERO-MEAN WHITE GAUSSIAN NOISE TO H*XS TO                ***00920
C     GENERATE A NOISY VERCTOR OF MEASUREMENTS Z.                       ***00930
C                                                                       ***00940
C                                                                       ***00950
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI       ***00960
```

```
      COMMON EI(5,5),Q(5,5),G(5,5),PKK(5,5),GAMMA(5,5),COVW(5,5),
     1TEMP(5,5),TEMP1(5,5),TEMP2(5,5),H(5,5),PKKM1(5,5),R(5,5),PHI(5,5),
     2VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),
     3GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),SIGW(5),X(5),
     4SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VTMP(5),Z(5),V(5),SIGV(5),
     5XHATZ(5),C(6),IEXT,IPHI,IH,
     6N,NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,M
C
      VP=1640.
      A=-1.
      P=3.1416
      CALL SNORM(IV,V,M)
C
      V(1)=SIGV(1)*V(1)
      V(2)=SIGV(2)*V(2)
C
      Z(1)=XS(4,K)-ATAN(A*XS(1,K)/XS(2,K))
C
      Z(1)=Z(1)+V(1)
      IF(XS(2,K).LT.0.) Z(1)=Z(1)-P
C
      R1=(XS(1,K)**2)+(XS(2,K)**2)
      R1=SQRT(R1)
C
      Z(2)=VP+(XS(3,K)*XS(2,K))/R1
      Z(2)=(XS(5,K)*VP)/Z(2)
      Z(2)=Z(2)+V(2)
      RETURN
      END
      SUBROUTINE TRACK                                                  ***02360
C     IF TRACK IS TO BE GENERATED ON-LINE IT IS DONE IN THIS SUBROUTINE ***02370
C     IN THE DEFAULT OPTION (ITRK.EQ.0) THE TRACK IS GENERATED          ***02380
C     FROM THE STANDARD LINEAR DIFFERENCE EQUATION                      ***02390
C                                                                       ***02400
C     X(K+1)=PHI*X(K)+GAMMA*W(K)                                        ***02410
C                                                                       ***02420
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI       ***02430
      COMMON EI(5,5),Q(5,5),G(5,5),PKK(5,5),GAMMA(5,5),COVW(5,5),
     1TEMP(5,5),TEMP1(5,5),TEMP2(5,5),H(5,5),PKKM1(5,5),R(5,5),PHI(5,5),
     2VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),
     3GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),SIGW(5),X(5),
     4SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VTMP(5),Z(5),V(5),SIGV(5),
     5XHATZ(5),C(6),IEXT,IPHI,IH,
     6N,NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,M
      DIMENSION W(5)
C     ITRK NE.0 OR 1 -- SEVERAL TRACKS GENERATED,BUT NOT FROM STD.      ***02520
C                       LINEAR EQS.                                     ***02530
C          = 0 -- SEVERAL TRACKS GENERATED FROM STD LINEAR EQS          ***02540
```

```
C          = 1 -- ONLY ONE TRACK IS USED                                         ***02550
        IF(ITRK.NE.0) GO TO 100                                                  ***02560
        CALL SNORM(IW,W,IN)
C       CONVERT EACH N(0,1) R.V. TO N(0,SIGW(I)) R.V.                            ***02580
        DO 1 I=1,IN                                                              ***02590
      1 W(I)=SIGW(I)*W(I)                                                        ***02600
        DO 3 I=1,N                                                               ***02610
      3 X(I)=XS(I,K)                                                             ***02620
        CALL VPROD(GAMMAS,W,N,IN,W)                                              ***02630
        CALL VPROD(PHIS,X,N,N,VTMP)                                              ***02640
        CALL VADD(VTMP,W,N,VTMP)                                                 ***02650
        DO 2 I=1,N                                                               ***02660
      2 XS(I,K+1)=VTMP(I)                                                        ***02670
C       NEW VALUE OF X HAS BEEN COMPUTED AND STORED IN THE ARRAY XS              ***02680
        RETURN                                                                   ***02740
    100 IF(ITRK.NE.1) GO TO 200                                                  ***02750
C        IF(ITRK.EQ.1) THE USER MUST INSERT HERE THE STATEMENTS REQUIRED         ***02760
C       TO GENERATE A SINGLE TRAJECTORY AND STORE IT IN THE ARRAY                ***02770
C       XS(I,K),I=1,N,K=2,NSAM (NOTE THAT IF A SINGLE TRAJECTORY IS TO BE        ***02780
C       GENERATED, THE INITIAL CONDITION HAS BEEN READ IN AND STORED             ***02790
C       IN XS(I,1),I=1,N)                                                        ***02800
C
C
        DO 5 K=2,NSAM
C
        DO 4 I=1,N
      4 X(I)=XS(I,K-1)
C
        CALL VPROD(PHIS,X,N,N,X)
        DO 5 I=1,N
        XS(I,K)=X(I)
      5 CONTINUE
        RETURN                                                                   ***02810
    200 CONTINUE                                                                 ***02820
C       IF THIS POINT IS REACHED, ITRK NOT EQUAL 0 OR 1 INDICATING THAT          ***02830
C       SEVERAL TRACKS ARE TO BE GENERATED, BUT NOT BY USING THE STD.            ***02840
C       LINEAR DIFFERENCE EQS..THE USER MUST SUPPLY THE APPROPRIATE              ***02850
C       STATEMENTS HERE.                                                         ***02860
        RETURN                                                                   ***02870
        END                                                                      ***02880
        SUBROUTINE XPRED
        REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI              MCSP0740
        COMMON EI(5,5),Q(5,5),G(5,5),PKK(5,5),GAMMA(5,5),COVW(5,5),
       1TEMP(5,5),TEMP1(5,5),TEMP2(5,5),H(5,5),PKKM1(5,5),R(5,5),PHI(5,5),
       2VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),
       3GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),SIGW(5),X(5),
       4SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VTMP(5),Z(5),V(5),SIGV(5),
       5XHATZ(5),C(6),IEXT,IPHI,IH,
```

```
     6N,NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,M
C
C     THIS SUBROUTINE CALCULATES THE STATE PREDICTION
C     MATRIX XHAT(K+1/K)=A(XHAT(K/K),U(K),K)
C     AND STORES IN THE ARRAY XHKKM1(I)
C
C
      CALL VPROD(PHIS,XHKK,N,N,XHKKM1)
      RETURN
      END
      SUBROUTINE CK
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI        MCSP0878
      COMMON EI(5,5),Q(5,5),G(5,5),PKK(5,5),GAMMA(5,5),COVW(5,5),
     1TEMP(5,5),TEMP1(5,5),TEMP2(5,5),H(5,5),PKKM1(5,5),R(5,5),PHI(5,5),
     2VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),
     3GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),SIGW(5),X(5),
     4SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VTMP(5),Z(5),V(5),SIGV(5),
     5XHATZ(5),C(6),IEXT,IPHI,IH,
     6N,NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,M
C
C     THIS SUBROUTINE CALCULATES THE CURRENT VALUE OFMATRIX C(X(K/K-1))
C
C
      A=-1.
      P=3.1416
      VP=1640.
      IF(Z(2).LT.500.) GO TO 1
      IF(XHKKM1(2).GT.0.) XHKKM1(2)=-XHKKM1(2)
      GO TO 2
    1 IF(XHKKM1(2).LT.0.) XHKKM1(2)=-XHKKM1(2)
    2 C(1)=XHKKM1(4)-ATAN(A*XHKKM1(1)/XHKKM1(2))
      IF(XHKKM1(2).LT.0.) C(1)=C(1)-P
      R1=(XHKKM1(1)**2)+(XHKKM1(2)**2)
      R1=SQRT(R1)
      C(2)=VP+(XHKKM1(3)*XHKKM1(2))/R1
      C(2)=(VP*XHKKM1(5))/C(2)
      RETURN
      END
      SUBROUTINE AK
C
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI
      COMMON EI(5,5),Q(5,5),G(5,5),PKK(5,5),GAMMA(5,5),COVW(5,5),
     1TEMP(5,5),TEMP1(5,5),TEMP2(5,5),H(5,5),PKKM1(5,5),R(5,5),PHI(5,5),
     2VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),
     3GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),SIGW(5),X(5),
     4SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VTMP(5),Z(5),V(5),SIGV(5),
     5XHATZ(5),C(6),IEXT,IPHI,IH,
     6N,NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,M
```

```
C     THIS SUBROUTINE CALCULATES THE A(K)=PHI(I,J) BY USING XHAT(K/K)
C     FOR EXTENDED KALMAN FILTER
C
      REAL*8 A(4)
C
C
      DO 1 I=1,N
C
    1 A(I)=XHKK(I)
C
      RETURN
      END
      SUBROUTINE HKK
C
C     THIS SUBROUTINE CALCULATES THE CURRENT LINEARIZED MEASUREMENT MATRIX
C     H(K) BY USING X(K/K-1)=XHKKM1(I)
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI
      COMMON EI(5,5),Q(5,5),G(5,5),PKK(5,5),GAMMA(5,5),COVW(5,5),
     1TEMP(5,5),TEMP1(5,5),TEMP2(5,5),H(5,5),PKKM1(5,5),R(5,5),PHI(5,5),
     2VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),
     3GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),SIGW(5),X(5),
     4SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VTMP(5),Z(5),V(5),SIGV(5),
     5XHATZ(5),C(6),IEXT,IPHI,IH,
     6N,NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,M
      REAL*8 A,VP,B(5),R1,F
      A=-1.DO
      VP=1640.DO
      DO 1 I=1,N
    1 B(I)=XHKKM1(I)
C
      R1=(B(1)**2)+(B(2)**2)
      R1=DSQRT(R1)
C
      F=VP+(B(3)*B(2))/R1
      F=(B(5)*VP)/F
C
      H(1,1)=-(A*B(2))/(R1**2)
      H(1,2)=(A*B(1))/(R1**2)
      H(1,3)=0.DO
      H(1,4)=1.DO
      H(1,5)=0.DO
C
      H(2,1)=(F**2)*B(3)*B(1)*B(2)
      H(2,1)=H(2,1)/(B(5)*VP*(R1**3))
      H(2,2)=-((F**2)*B(3)*(B(1)**2))
      H(2,2)=H(2,2)/((R1**3)*VP*B(5))
      H(2,3)=-((F**2)*B(2))
      H(2,3)=H(2,3)/(B(5)*VP*R1)
```

```
      H(2,4)=0.D0
      H(2,5)=F/B(5)
C
C
      RETURN
      END
```

BIBLIOGRAPHY

1. D. E. Kirk, State Estimators and Predictors for Fire Control Systems, Naval Postgraduate School report NPS-52K174101, Monterey, California, October 1974.

2. Davenport, Probability and Random Processes, McGraw-Hill, 1970, p. 250-251.

3. Jazwinski, Stocastic Processes and Filtering Theory, Academic Press, 1970, p. 39-41.

4. G. W. Mitschang, An Application of Nonlinear Filtering Theory to Passive Target Location and Tracking, Ph.D. Thesis, Naval Postgraduate School, Monterey, California, 1974.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Professor D. E. Kirk, Code 52Ki Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
4.	Professor H. A. Titus, Code 52Ts Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
5.	Bogazici Universitesi Arnavutkoy, Istanbul, Turkey	1
6.	Ortadogu Teknik Universitesi Ankara, Turkey	1
7.	Istanbul Teknik Universitesi Istanbul, Turkey	1
8.	Toshiaki Iida 153 Terayama Hatano-City Kanagawa, Japan	1
9.	LCdr. Marcilio Bpa Cunha Naval Postgraduate School S.M.C. 2379 Monterey, California 93940	1
10.	Lt. H. O. Tasdelen Denizalti Filosu, Golecuk Kocaeli, Turkey	2